1-cryptographic-effectiveness.md 10.4 KB

Discussion Block 1 - Effectiveness of Cryptography

Thesis

The first two questions of the questionnaire were about the effectiveness of cryptography (the specification of the full protocol as well as the underlying building-blocks). From the replies to the questionnaire, we conclude that the effectiveness of cryptography strongly hinges on its correct use. Despite the challenges at getting it right, there is no way around cryptography when it comes to internet voting.

Areas of concern related to wrong usage of cryptography:

1. Flawed implementation (this can involve errors in design-documents or code but also bad choices at instantiating from higher-level abstractions or when combining multiple building-blocks);
2. Building-blocks particularly risk being flawed if they are not taken from widely ac-cepted standards and if they are modified;
3. Particularly zero-knowledge proofs and verifiable mix-nets for voting are complex and generally defined in research-papers and not taken up by widely accepted standards. Due to the generally high abstraction level, important considerations (e.g. setup, surrounding environment) are often not made explicit;
4. Building blocks or their mode of operation might not address the real-world needs (trust assumptions, attacker capabilities) sufficiently unless carefully chosen;
5. Regarding the real-world needs: Quantum computers or advances in cryptanalysis may at some point subvert the soundness of today's standard building blocks;

The following risk-limiting measures are effective:

6. Involve experts from cryptography at all stages that underlie the effectiveness of cryptography. Particularly ensure vast and continuous scrutiny. Furthermore, the latest advances in cryptanalysis must be taken into account;
7. Security proofs and their verification;
8. Formal methods and automated proof-checking reduce the risk of a flawed proof and therefore of a flaw in the protocol;
9. Closely relate choices to real-world needs, e.g. security proofs need to be checked against these needs.
10. Work toward rigorous standardization of building blocks of use both in e-voting systems and broader-interest communities (such as zero-knowledge proofs, verifiable mix-nets, and bulletin board protocols), and build on these standards once available.

Related Questions

The related questions are labelled Block-1.

Individual links to related questions

- Block 1 Discussion A - Thesis Validation
- Block 1 Discussion B - Number of Reviewers
- Block 1 Discussion C - Abstraction Level of Specification
- Block 1 Discussion D - Reviews of Operations
- Block 1 Discussion E - Formal Methods
- Block 1 Discussion F - Breaking Cryptography

Questionnaire

The thesis is based on the questions 2.1.1 and 2.1.2 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.1.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.1.2	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 1A - Thesis Validation (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness

Questions

- Do you agree with these conclusions?
- Is there anything important that would not fall under these points?

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 01, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

Christian Folini @christian.folini changed the description 7 months ago

David Basin @David.Basin · 7 months ago Developer

Yes: agreement with the summary and its coverage.

Vanessa Teague @Vanessa.Teague · 7 months ago Developer

I also agree that this is a good summary.

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you for breaking the ice, @David.Basin and @Vanessa.Teague.

We're preparing the next discussion block with more questions. But before we launch it, we'd like to at least confirm the thesis (or hear of fundamental objections, if we misread the responses to the questionnaire). The number of reviewers (Discussion B) is quite a central question in many ways too.

So we ask all participants to think about this and either confirm or reject the thesis.

Namely the opinion of the following experts matters to the dialogue: @Srdjan.Capkun, @Eric.Dubuis, @Philipp.Locher, @Reto.Koenig, @Rolf.Haenni, @Olivier.Pereira, @Carsten.Schuermann, @Bryan.Ford, @Bogdan.Warinschi and @Ulrich.Ultes-Nitsche.

Florian Egloff @Florian.Egloff · 7 months ago Developer

I defer to the cryptographers involved to answer this block of questions.

The only thought that came to mind when reading through the thesis, is that for (5) i.e. the erosion of cryptographic soundness due to improvements in cryptanalysis over time, one would need a mitigation that is not made explicit in (6) (i.e. what are all relevant stages?).

In my opinion, it would have to be explicit that cryptographic expertise needs to be consulted over time and not just at the design stage. In other words, the 'relevant stages' need to be made explicit.

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you for this reminder Florian. Question 3.5 in the questionnaire is focusing on this aspect. We plan to return on this question during the dialogue, since there is not universal agreement among the experts so far. Is this OK for you or do you think it should go into this thesis / statement as well?

- 3.5 Question
- 3.5 Summary
- 3.5 All Responses

Collapse replies

Florian Egloff @Florian.Egloff · 7 months ago Developer

Yes, this is fine for me.

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you.

Srdjan Capkun @Srdjan.Capkun · 7 months ago Developer

This looks good to me. Main points are well covered.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

I also agree. This is a good summary. My only comment is about 3. Crypto is not only complex and difficult to grasp, but relies on residual assumptions, for example, probabilistic poly-time adversaries, the hardness of an underlying mathematical problem, or that hash functions can be used as random oracles. Is this included in point 4. trust assumptions?

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

I think the summary captures well where building blocks fail (poor implementation, non-ideal fit, improvements in cryptanalysis) but does not mention (except perhaps implicitly) that the main problem is that of protocol design. Putting even the strongest building blocks together can easily lead to issues.

Nonetheless, the mitigating actions are designed to tackle this issue so maybe the omission is minor for the discussion (but should be included in any document that we produce).

Christian Folini @christian.folini mentioned in issue #9 (closed) 7 months ago

Olivier Pereira @Olivier.Pereira · 7 months ago Developer

Hi, All, I agree with this summary as well! Two possible observations:

- I would find it important to make sure that we do not give the feeling that crypto is just pointless because it can fail in so many ways. It is definitely hard to get it right, for all the reasons pointed here, but it is also the best/only tool we have in order to address many of the problems that are relevant here.
- I wonder if "Involve experts from cryptography at all relevant stages" might be too vague (what are the relevant stages?). In particular, and related to @Carsten.Schuermann 's point, we may have a perfect protocol design w.r.t. a spec, with machine checked proofs, but the underlying computational assumptions may become problematic at some point (e.g., because of advances in algorithmic number theory, ...), or someone may notice that the security model/definition does not capture some relevant threats, etc. So, I wonder if it would make sense to already suggest that the "relevant stages" are somehow "all the time", including after initial validation and deployment.

Collapse replies

Florian Egloff @Florian.Egloff · 7 months ago Developer

I agree with having to address the vagueness in ‘relevant stages’. This is also the root cause for my comment above.

Reto Koenig @Reto.Koenig · 7 months ago Developer

We (e-voting group BFH) agree with the summary. However, we would like to emphasise that erosion of cryptographic soundness is a problem that can be tackled over time (adapting the cryptographic parameters or even cryptographic-primitives), whereas the erosion of privacy cannot be tackled once voting data is made available. Be it via open bulletin board required for universal verifiability or a leak at providers side.

Collapse replies

Oliver Spycher @oliver.spycher · 7 months ago Maintainer

Reto, you say that the erosion of privacy cannot be tackled once voting data is made available. Can you relate your statement to schemes known from the literature that aim at information-theoretically secure privacy guarantees (“everlasting privacy”)?

Reto Koenig @Reto.Koenig · 7 months ago Developer

Apart from approaches using post-quantum cryptography, a variety of possibilities exist to mitigate the long-term privacy problem or even aim at **unconditional everlasting privacy**. One approach is to link vote encryptions with perfectly hiding commitments, using techniques such as **commitment consistent encryptions**, and to define the universal verification process based on these commitments (CPP13,DVA12,ACKR13). Another approach we have studied is based on **set membership proofs in zero-knowledge** (LH15,LH16,LHK16).

We are well aware that these approaches all come with some additional assumptions or limitations, e.g., the existence of an anonymous channel during vote submission, or the ability to scale linear with the number of voters.

Olivier Pereira @Olivier.Pereira · 7 months ago Developer

Hi, I guess that we can make a distinction here between an attacker who sees a public bulletin board, and an attacker who monitors Internet connections. We can hope for everlasting privacy in the bulletin board case (using the techniques mentioned by @Reto.Koenig -- and this can be done in linear time), but not when we face someone who eavesdrops on Internet connections, since the information on the vote (in the information theoretic sense) needs to travel through the Internet channel (if we do not send it, we cannot tally), and can therefore only be protected in a computational way.

Christian Folini @christian.folini · 7 months ago Maintainer

Thanks for these valuable comments on the thesis.

I think you all agree with the general direction of the thesis. However, we are preparing a slightly adjusted wording to take up your feedback.

Oliver Spycher @oliver.spycher · 7 months ago Maintainer

Thanks @Reto.Koenig and @Olivier.Pereira for clarifying

Bryan Ford @Bryan.Ford · 7 months ago Developer

The thesis and summaries look good to me overall, but I think are missing coverage of one or two key points regarding standards that I think emerge from the opinions and discussion so far. In particular, one key point is that building-block protocols like TLS have become reasonably secure in part because receive broad interest and scrutiny far beyond any one application domain. Other building-blocks used in E-voting, such as zero-knowledge proofs, verifiable shuffles, and bulletin boards, in principle are of much broader interest beyond E-voting and in principle "could" be embodied in standards receiving wider scrutiny like TLS does, but this haven't happened yet for a variety of reasons.

First, I would suggest refining thesis point 2 a bit, to something like:

"2. Building-blocks have a higher risk of being flawed if they are not taken from widely accepted standards, and if they modify the standards, or if the standards are developed and scrutinized by a narrower community (e.g., the E-voting community only, or even worse the small community around one particular E-voting system) as opposed to a much broader community (e.g., all the companies who use AES, ElGamal, TLS)."

This modified thesis point could also be split into a 2a and a 2b (the latter being about the narrowness/breadth of the community and the scrutiny a building block or standard receives).

I would similarly suggest splitting thesis point 3 into two and generalizing the first part a bit, something like this (too long I know, sorry):

"3a. Many building-blocks used in voting systems are complex, are defined only in research papers appealing only to a narrow audience (e.g., E-voting community only), and are not taken up by standards accepted and scrutinized by the broader communities having a common interest in these building blocks. Examples of such building blocks include zero-knowledge proofs, verifiable mix-nets, and bulletin boards (aka tamper-evident logs, distributed ledgers, or blockchains), for which there is broad interest beyond e-voting that could in principle lead to widely-scrutinized standards akin to TLS, but this standardization has not happened yet.

3b. Due to the generally high abstraction level, important considerations (e.g. setup, surrounding environment) are often not made explicit"

And finally, I would add a standards-related measure to the "risk-limiting measures" list below:

"10?. Work toward rigorous standardization of building blocks of use both in e-voting systems and broader-interest communities (such as zero-knowledge proofs, verifiable mix-nets, and bulletin board protocols), and build on these standards once available."

Vanessa Teague @Vanessa.Teague · 7 months ago Developer

Can we kindly have a separate discussion on whether the term 'bulletin board' is equivalent to the terms "tamper-evident logs, distributed ledgers, or blockchains"? I would say no, and I feel the discussion is possibly relevant enough to include in this dialog, but not important enough to get diverted by here.

Collapse replies

Christian Folini @christian.folini · 7 months ago Maintainer

Yes.

A fairly extensive discussion block on "bulletin boards" is clearly in the book. It will be interesting to hear all the expert opinions on the adjacent terms you named.

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago Developer

agree with the thesis.

Eric Dubuis @Eric.Dubuis · 7 months ago Developer

I aggree with the summary. However, I suggest to add another, explicit point to the risk-limiting measures, something like "Ongoing, public scrutiny on all levels of implementation (software and even hardware)".

Christian Folini @christian.folini changed title from **Discussion A - Thesis Validation (Block 1 - Cryptographic Effectiveness)** to **Discussion 1A - Thesis Validation (Block 1 - Cryptographic Effectiveness)** 6 months ago

Bryan Ford @Bryan.Ford mentioned in issue #57 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

The thesis document has been updated following your feedback (yes, this took a while ...).

Here is the `diff`.

If you agree with this updated text, then I am proposing the following summary for this question here:

Do you agree with these conclusions?

Is there anything important that would not fall under these points?

The conclusions are correct and all important aspects have been covered.

If you are still not OK with the conclusions, then please let me know and we'll work towards another update. If I do not hear from you, I'm going to assume everybody agrees now.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

I have not heard any additional feedback. Given most people agreed to the initial thesis, I am assuming content with the slightly updated thesis and conclude this discussion.

Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 1B - Number of Reviewers (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness, thesis points (2),(3),(6),(7)

Questions

If one independent cryptographer is mandated to verify the protocol specification and the proofs against the requirements, how likely is it that the protocol specification does not meet the requirements?

Assuming that the documentation is published, is there significant added value of appointing two independent cryptographers instead of one, and if so, should they work jointly or write separate reports?

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 02, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

David Basin @David.Basin · 7 months ago Developer

Depends on the skills of the cryptographer and the nature of the proofs. For example, if the proofs are machine-checked (symbolic or game-hopping) then the majority of the work is done by the theorem-prover/model checker. As a concrete example, game-hopping proofs are machine checked from foundations built onto of probability theory and probabilistic programs in https://eprint.iacr.org/2017/753 A cryptographer charged with checking the proofs would mainly have to check the statement of what was proven, but not the proof steps themselves. For proofs on paper the situation is quite different as humans make mistakes (prover and checker) and moreover the steps are often rather high-level, omitting details. Here the skills of the prover and the expert are relevant to determining whether the omissions are acceptable. In this setting having 2 independent checkers is desirable. Note that for model-checking, where substantial abstractions and simplifications are being made, there are also details to be "filled in", so for the same reason a second expert could be helpful.

Christian Folini @christian.folini mentioned in issue #2 (closed) 7 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 7 months ago Developer

In domains other than cryptography, multiple independant reviews are the norm if correctness is critical. I believe at least two reviewers are needed to produce independent reports, though each should be able to comment on the report of the other.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

As David says, it depends on the skills of the cryptographer. Experience shows that it is very difficult to produce proofs without bugs -- and most cryptographers don't use formal methods. Machines can check the proofs and find the mistakes, but they raise to new challenges: 1. Does the model reflect the theory correctly (mistakes in representing the underlying algorithms, and mathematical assumptions, can lead to correct machine checked proofs about a different system)? 2. Does the formalized theorem really represent the mathematical fact to be proven (mistakes in the representation of the theorem might mean that we are accidentally proving something else)? Even with modern tools, it is easy to prove something unintended.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

I'll limit these comments to cryptographic proofs (and not formal methods ones which were already covered).

A single skilled expert would most likely suffice to ensure that the formal models used in proofs capture the requirements. The complexity of checking the proofs is of a different order of magnitude: at the moment there is only little work which can be used to produce cryptographic machine checkable proofs at the scale required by electronic voting, and cryptographic proofs are (overwhelmingly) pen and paper. Even following best practices, these proofs will be highly complex and to some extent incomplete. In my view having such proofs checked by two independent experts it a rather minimal requirement. The reports should be written independently, but follow-up discussions between the experts would also be beneficial.

Christian Folini @christian.folini mentioned in issue #6 (closed) 7 months ago

Reto Koenig @Reto.Koenig · 7 months ago Developer

We (e-voting group BFH) agree with the other expert. In our view the biggest challenge is to find the correct abstraction level for the proof. Thus, the more reviewers that can check the given proofs, the more unlikely it becomes, that important aspects have been omitted for the proof generation.

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago Developer

I agree with the previous opinions. One or two experts are not sufficient for a detailed review. An optimal combination of machine and human, carried out independently. A procedure in several steps makes sense where in a first step several parties (individuals or groups) check the functions for themselves, then in a dialogue. A corresponding track of the persons is assumed.

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you for your contributions in this discussion.

Let me wrap it up and you tell me if I got it right:

If one independent cryptographer is mandated to verify the protocol specification and the proofs against the requirements, how likely is it that the protocol specification does not meet the requirements?

Answer: There are two things to be analyzed: (1) that the right statement is proven and (2) that the proof is correct. (1) is nontrivial both in the computational and the symbolic setting. (2) is nontrivial for the computational setting and trivial in the symbolic setting, at least when the proof is machine checked. Hence for both (1) and (2) in the computational setting, two cryptographers are better than one to avoid errors in the verification.

The fact, that (1) is nontrivial may be surprising, but it is a particular characteristic of cryptography.

Machine-checking proofs is helpful, but it comes with problems of its own and the tools existing today are not able to cover all areas of the cryptography used for an online voting system (Also see discussion E).

Assuming that the documentation is published, is there significant added value of appointing two independent cryptographers instead of one, and if so, should they work jointly or write separate reports?

Yes, there is significant value. They should work independently, but discussing the specifications or findings in a 2nd step or an iterative process would be beneficial. Two independent cryptographers is the minimum. Yet, additional reviewers would still be beneficial.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

[EDIT]: Incorporated the reworded proposal by @David.Basin below into the 1st answer which is much more to the point now. This 1st answer also covers the 2nd question to a wide extent, but I am keeping the 2nd answer since it also responds to the question whether they should work independently or not.

Edited by Christian Folini 7 months ago

Bryan Ford @Bryan.Ford · 7 months ago Developer

Your summary looks good by my reading!

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you Bryan.

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago Developer

works for me as well.

Collapse replies

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you Tobias.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

To clarify one hidden step in the summary.

The security analysis of a protocol specification goes through two stages. a) specify a formal security model (trust assumptions, execution model, adversary model, when does the adversary break the protocol, potentially abstractions of protocol steps, etc) and b) mathematically prove that the specification satisfies the model (i.e. no adversary from the class considered in the model can break the protocol under some clearly identified assumptions).

To verify that such an analysis implies that the specification meets the official requirements there are again two steps: i) check that the security model used in the analysis captures/reflects the requirements and ii) check that the proof is valid.

In my view, in the summary above the first step refers to step i), indirectly. I think it should be more explicit. In particular, it can justify a requirement that a protocol analysis includes an (unavoidably informal) justification on how the security model captures the requirements; the verification of the analysis by independent experts would need to confirm that this is indeed the case.

David Basin @David.Basin · 7 months ago Developer

I second what @Bogdan.Warinschi said. Moreover, just to be clear, one must also "go down" at some point and prove the execution model is a proper abstraction of the actual implementation. If this isn't the case, the proof that the protocol specification (encompassing all that Bogdan said) meets its specification is worthless as it is about the wrong system.

Regarding 2 cryptographers: note that the field is surprisingly specialized. Some cryptographers can judge either computational or symbolic proofs, but not both.

Edited by David Basin 7 months ago

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you @Bogdan.Warinschi and @David.Basin . I see your point, but I have a hard time integrating it properly into my summary. Could you propose a rewording that covers your statements as well?

I tried to keep the wording abstract but I realize that we need to descend into the particulars. And if you propose a wording, then we can avoid unnecessary ping-pong.

David Basin @David.Basin · 7 months ago Developer

I appreciate the need to keep things abstract. Very abstractly: there are two things to be analyzed: (1) that the right statement is proven and (2) that the proof is correct. (1) is nontrivial both in the computational and the symbolic setting. (2) is nontrivial for the computational setting and trivial in the symbolic setting, at least when the proof is machine checked. Hence for both (1) and (2) in the computational setting, two cryptographers are better than one.

The statement that (1) is nontrivial may be surprising to some. We are used to thinking of checking theorem statements as being relatively straightforward. Here it is not, for the reasons explained above.

Collapse replies

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you David. I have adopted your proposal and integrated it into my summary above. Much clearer now.

Christian Folini @christian.folini added Last-Call label 7 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Christian Folini @christian.folini mentioned in issue #14 (closed) 6 months ago

Christian Folini @christian.folini changed title from **Discussion B - Number of Reviewers (Block 1 - Cryptographic Effectiveness)** to **Discussion 1B - Number of Reviewers (Block 1 - Cryptographic Effectiveness)** 6 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 1C - Abstraction Level of Specification (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness, thesis points (3), (6), (7)

Questions

There is a protocol-specification at the top (the exposition which the security proofs relate to), the code at the bottom and specifications in between. Is it possible to express in words what maximum level of abstraction of the protocol specification seems admissible without introducing a significant risk that the lower level specification does not instantiate the protocol specification appropriately?

(How specific should the security-proven protocol-specification be e.g. with regard to setup procedures, parameter generation?)

Assume that the lower level specification is scrutinized by the same cryptographer as the protocol specification.

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 03, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

David Basin @David.Basin · 7 months ago Developer

From the formal methods perspective, the question boils down to: can one construct a series of specifications (also called models) S_1, ... S_k starting with a very high-level specification of the protocol where properties are established going down to a low-level design or even code specification, where the i+1st model "instantiates" or (formally speaking) "refines" the ith model? This is known as development by refinement and it is applicable to security protocols. See, for example: http://people.inf.ethz.ch/basin/pubs/jcs18.pdf . The question then becomes how difficult is it to relate adjacent models: formally construct a refinement relationship between them. The answer is: one cannot easily express this in words. If the gap between models is small, the "refinement relation" (also called "simulation relation") is easy to construct. If it is large, more work must be taken to come up with the refinement relation and to prove that it is a refinement relation. (Of course: it must be proved!)

If one interprets "refinement" or "instantiation" in some informal way, then it becomes much easier, of course, to eyeball the specifications and assert that they look reasonable, but this assertion would have no formal meaning and there would be no guarantees about the correctness of the actual code.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

What David says. One important thing to consider, in addition, is the implementation language and what kind of guarantees it can and should give. Everyone would agree, that assembly or even C do not give many guarantees in terms of memory safety, strongly typed languages on the other hand do: Programs won't crash with a segmentation fault. If the implementation language provides information flow control, for example, the refinements can take advantage of these guarantees.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

I also agree with David: it is difficult to prescribe a specification level which will ensure that i) it is feasible to reason about the specification and ii) can be rigorously related to lower levels of specification (or code).

One additional comment to Carsten's observation: based on some of my current experience (I'm at the moment in a start-up where part of my work is to ensure that code implements our specifications), code quality and programming language used are also particularly important to help the process. Even if the specification is detailed (low level), the implementation may still use language

specific features/design patterns which make establishing the desired link complex -- I often have multiple back and forth with the engineers on precisely this issue.

Reto Koenig @Reto.Koenig · 7 months ago Developer

In our (e-voging group BFH) opinion, it is an absolute must, that the implementer (programmer) as well as the prover of the protocol-specification do not have (to / the freedom to) 'interpret' any mathematical / cryptographic aspect. Hence, it must be deterministically specified (aka down to the level of pseudo-code), how to represent values, how to concatenate values, how to hash values, how to sign values, how to gain randomness etc.

This naturally results in the specification that has to be provided down to pseudo-code that must be the only base for proof and implementation.

I.E. The 'specification in between' serving the only purpose of defining independent implementation specifics file format or APIs e.g. JSON et al, does not bear any cryptographic relevant aspects. On the other hand, the protocol specification is completely oblivious to any of the programming languages used for implementation.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

I'd like to amend a bit my statement (which was probably too much influenced by dealing with non-cryptographic specifications): as @Reto.Koenig says, for all cryptographic aspects specification can and should be at pseudocode level detailing the algorithms used, the sizes of the keys, nonces.

In fact, I would like to highlight the ChVote specification by the BHF group (https://eprint.iacr.org/2017/325.pdf) as an excellent example of how a specification could/should look like. I was involved in its analysis (https://eprint.iacr.org/2018/1052) and our job was certainly helped by how thorough (and well structured) that specification is. In particular, the zk proofs, the mix-nets, OT protocols, etc were described in detailed pseudocode and I can imagine that this would leave less room for errors and interpretation.

Conversely, and this remark reflects our experience with the security analysis for the ChVote specification, devising the proofs, both symbolic and computational ones, we needed to go in the reverse direction, i.e. consider increasingly more abstract specifications to the point where the proofs become manageable. These abstraction steps need to be closely scrutinized, especially for the more abstract symbolic models.

David Basin @David.Basin · 7 months ago Developer

The limiting case of what Bogdan is referring to is when the specification is extremely concrete, namely a ("reference") implementation in a programming language. The abstraction step to a verifiable model would then be huge and the refinement relation relating the code to the more abstract model extremely complex. In the end, whether you start very abstract or very concrete you have to relate the levels.

Incidentally, in my group and the group of Peter Mueller at ETH Zurich we are working on a methodology to bridge development by refinement (top down) with code-level verification using tools of Peter (Viper/Nagini), with a formal bridge between (relatively concretely) abstract models and specifications of (Go/Python/Rust) code in an I/O separation logic. So there are methodologies out there to combine top-down and bottom-up development and verification. But this is still research.

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago Developer

agree with the statements here especially from @Reto.Koenig . no additional comments.

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you for your comments in this thread.

I'm understanding that you see a need to have several levels of specifications to go from the protocol to the code and that the specification that the developers / engineers use, should not leave any room for interpretation, especially not for the cryptographic part of the software where the low level specification should include pseudo-code. The width of the steps between the levels of specifications are very important, because it is very hard to prove a so called refinement relationships between different levels of specifications if the gap is too wide.

Choice of implementation language also plays a major role, as does the quality of the code.

Before we close this discussion, there is a part of the question that has not been explicitly addressed by your responses, though: The operational aspects of the specification:

(How specific should the security-proven protocol-specification be e.g. with regard to setup procedures, parameter generation?)

I suppose you want these procedures to be specified in a very detailed way as well, to make sure there is no room for interpretations

/ misunderstandings. Correct?

Christian Folini @christian.folini changed title from **Discussion C - Abstraction Level of Specification (Block 1 - Cryptographic Effectiveness)** to **Discussion 1C - Abstraction Level of Specification (Block 1 - Cryptographic Effectiveness)** 6 months ago

Christian Folini @christian.folini mentioned in issue #18 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

I have not heard any feedback to my interpretation above. So I take it I'm not too far off.

Let's try a summary:

It is difficult to describe appropriate levels for different types of specifications.

The top specification of the protocol has to be a verifiable model that is abstract enough to allow reasoning about it.

The bottom specification has to be very detailed and include pseudo-code for every mathematical / cryptographic aspect in order to avoid any room for interpretation for the developer of the software as well as the people proving / checking it. Non-cryptographic aspects do not demand such a detailed specification.

There has to be a refinement relationship between the different levels of the specification down to the pseudo-code and eventually to the code level. This refinement relationship can only be established, when the step width is not too big. The ideal width is difficult to define and still a topic of research.

The choice of the implementation language plays an important role, since features like strong typing and flow control can be used in the low level specification. Language specific features and / or design patterns can make the review more difficult.

Setup procedures, parameter generation and everything that touches on the implementation of the cryptography has to be specified with no room for interpretation as well.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT: Dropped a statement and extended one after input by @David.Basin .] [EDIT: Typo]

Edited by Christian Folini 6 months ago

Christian Folini @christian.folini added Last-Call label 6 months ago

David Basin @David.Basin · 6 months ago Developer

I am in basic agreement but even with good specifications the proofs are not necessarily simple. Often one must come up with clever invariants and the refinement relations are not always trivial. An easy fix in your summary is just to delete the sentence "With a functioning ..."

Note too that in the ideal case the refinement goes all the way down to the code itself, not just pseudo-code. This is possible but requires considerable (theorem proving) work.

There is no free lunch: if you want to close all gaps, the specification and proof process is nontrivial, requires expertise, and is time consuming. One can stop at a higher level than code, but the guarantees are much weaker.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for your feedback David. In fact, the sentence that you propose to drop is based on your "If the gap between models is small, the 'refinement relation' (also called "simulation relation") is easy to construct."

Am I messing things up here or where does the contradiction come from? Could you please clarify?

Other than that, I'm perfectly OK to drop that statement. I just want to make sure I got it right.

As for your 2nd note, how about extending the following statement?

Now: "There has to be a refinement relationship between the different levels of the specification."

Extension: "There has to be a refinement relationship between the different levels of the specification down to the pseudo-code and eventually to the code level."

David Basin @David.Basin · 6 months ago Developer

Sorry I wasn't clear. If the refinement gap is small then the refinement relation is (relatively) easy to construct. However, you need a lot of them. In my group we once developed train controller software on a project with industry by refinement and the proof involved over 80 models from abstract to concrete. So constructing the refinement relation was simple but it was not a simple matter to go from top to bottom because (1) there were very many refinement steps and (2) sometimes, when there were problems going from the i-th to the i+1-st model it was because one of the earlier models wasn't sufficiently abstract enough, and hence there was some time-consuming back & forth, where earlier sequences of models had to be reformulated and proofs redone.

There was another point in the above but somewhat technical and it can be omitted (I mention it here for completeness): in establishing correctness you need both to establish refinement relations between models and also properties of the individual models, which are typically invariants. The invariants are usually not too hard to prove (induction over the model's transitions) but sometimes auxiliary invariants are needed and they are not always so easy to come up with.

Summarizing the above: the individual refinement steps, when small, can be simple to construct. However refinement proofs of the entire system, from high-level specifications to low-level models or code, are not necessarily simple. (Theorem proving takes work.)

I like your extension.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you David. Much clearer now. It is very helpful to have such a good description on this in this thread (for future reference).

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Christian Folini @christian.folini mentioned in issue #28 (closed) 6 months ago

Closed Opened 7 months ago by Christian Folini Reopen issue

Discussion 1D - Reviews of Operations (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness, thesis points (1),(6)

Question

Is it relevant to appoint cryptographers at scrutinizing not just the specification and the code, but also the operational procedures, e.g. defined for printing, tallying and operating the control-components and the verifier?

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues 0

Christian Folini @christian.folini changed due date to January 04, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

Christian Folini @christian.folini changed the description 7 months ago

David Basin @David.Basin · 7 months ago Developer

Sure cryptographers are smart people. :-) But seriously: if any of these "operational procedures" involve cryptographic operations (even relatively innocent looking ones like generation of randomness) then yes. And certainly when the operations involve decryption, mixing, checking ZKPs, etc.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

Yes. Cryptography doesn't solve the problem of security, it just moves it elsewhere, for example, the management of keys, independence of control components and mixers, access to true randomness. The way how operational procedures affect the security depends critically on the operational procedures being compatible with the assumptions cryptographers make, and they should therefore be involved.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

Yes -- especially on those operational aspects which motivate the trust model used in the models/analysis: e.g. key generation, randomness generation, multi-party computation.

Reto Koenig @Reto.Koenig · 7 months ago Developer

In our (e-voting group BFH) opinion, it is important that the operational procedures respect and cover the security model. Hence, this aspect has to be audited by security experts that have deep understanding especially on physical layer. E.g. how to treat the physical memory for the private key. How to ensure that no side channel is available to read out privat data.

Christian Folini @christian.folini mentioned in issue #7 (closed) 7 months ago

Christian Folini @christian.folini · 7 months ago Maintainer

Thank you for your contributions in this discussion.

I think we have consensus here. Let me sum it up:

Yes it is imporant to appoint cryptographers at scrutinizing the operational procedures wherever they are relevant to the cryptographic operations or the assumptions that the cryptographers make. It is also important that these expert have a deep understanding down to the physical layer.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

David Basin @David.Basin · 7 months ago Developer

Christian Folini @christian.folini added Last-Call label 7 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini mentioned in issue #14 (closed) 6 months ago

Christian Folini @christian.folini changed title from **Discussion D - Reviews of Operations (Block 1 - Cryptographic Effectiveness)** to **Discussion 1D - Reviews of Operations (Block 1 - Cryptographic Effectiveness)** 6 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 1E - Formal Methods (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness, thesis points (8)

Questions

The VEleS requires a security proof based on symbolic methods. Would the formal methods and automated proof-checking you would think of meet that requirement? Would the requirement need to be more precise as to ensure that effective methods are implemented? What is the added value of the formal methods you might have thought of

- in terms of additionally reducing the risk that the protocol specification does not satisfy the requirements; and
- in terms of efforts / costs?

Edited 6 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 05, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

Christian Folini @christian.folini changed the description 7 months ago

David Basin @David.Basin · 7 months ago Developer

The current setup has the advantage of requiring two kinds of proofs. Computational proofs offer the strongest guarantees possible today. But they are rarely machine-checked (for exceptions see: http://people.inf.ethz.ch/basin/pubs/jcrypto19.pdf). Hence checking the proofs can be error prone. Symbolic proofs can be automatically checked.

I would NOT legally mandate the use of a particular tool: over time tools come and go. For checking symbolic proofs though there are two state-of-the-art tools: Tamarin and Proverif. Using either of those should be fine.

Summarizing: symbolic proofs reduce the risk of many different kinds of protocol errors and also the risk that a human may accept an incorrect proof as a correct one. Such formal methods should unquestionably be used to evaluate the voting protocols used. History has shown cryptographic protocols, like those used for voting, are subtle and far too easy to get wrong without machine support to reason about all the ways that an attacker can interact with the system to defeat its security.

Note: proofs are time consuming (and therefore costly) to develop. But the alternative, a flawed voting system, is even more costly to fix in practice.

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

Machine checked proof would play a crucial role in reducing the gap between the requirements and the protocol specification: assuming that the framework is sound then one has to map the model used in the analysis to the requirement, and the abstraction used in the analysis (if any) to the specification.

A small addition to David's points:

Demanding a machine checked proof with respect to a *symbolic* model is not a tall order: one of the main benefit of such models is that they enable automation. Any of the state-of-the-art tools would provide similar guarantees and benefits.

A machine checked proof with respect to a *computational* model would be highly desirable. As I outlined at point B of the discussion such proofs have omissions, make unspecified assumptions and are often buggy. However, there are only a few tools

that support such proofs (I'd add Easycrypt and friends, and Cryptoverif to David's example) and it is unclear if they offer the full support required by a security proof in this space. A technical example: the forking lemma needed to reason about zero-knowledge proofs is at the moment not supported in the tools that I mentioned.

So, a *full* machine checked computational proof may be out of the reach of the state of the art. NB: one could still carry out a proof by relying on pen and paper proofs for some parts of the system and assume their security within the formal proof.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

Yes. I agree with with what David and Bogdan say. Computational proofs are desirable, but somewhat difficult and, without any formal support, easily prone to error. But the tools are simply not there yet. Symbolic proofs are better understood, especially the tools for mechanized proof. However, symbolic proofs also have limitations due to combinatorial space explosion.

Christian Folini @christian.folini · 7 months ago Maintainer

Please note, that several responses in the Discussion B (Number of Reviewers) of this discussion block also talk about machine-checking of proofs.

Christian Folini @christian.folini mentioned in issue #3 (closed) 7 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

Time to wrap this up. I propose you the following summary / response:

Checking proofs automatically brings very good assurance, as it is less prone to human errors. Symbolic proofs can and should be checked automatically, since this is well established. Computational proof used for online voting can not yet be checked automatically with the software existing today. Therefore we still need manual checking.

Manual checking is also imperative since the two different kind of analyses - computational and symbolic - tend to find different attacks resulting from different flaws: The computational analysis is strong with respect to the cryptographic functionality, the automated symbolic analysis is more detailed with respect to the protocol's control flow and the possible ways that adversaries can interact with interleaved protocol runs.

Security proofs and checking the security proofs is an ongoing process. The proofs are established with regards to security definitions and models that are constantly being reviewed and updated themselves. As time passes, it is quite plausible a definition does not capture some important security aspect and that a different proof becomes necessary.

The efforts and costs for these proofs and the checks are very high, but there is no way around it.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT: Incorporated feedback by @David.Basin and @Olivier.Pereira. New wording leans heavily on their wording.]

[EDIT: Another slight rewording based on @David.Basin 's proposals.]

[EDIT: Typo]

Edited by Christian Folini 6 months ago

Christian Folini @christian.folini added Last-Call label 6 months ago

David Basin @David.Basin · 6 months ago Developer

@christian.folini : there are good reasons have to both kinds of proofs which go beyond the question of "manual versus automatic" and concern coverage. A symbolic model and the associated analysis, while more abstract with respect to how cryptographic functionality is handled, is usually much more detailed than a computational one with respect to the protocol's control flow and all the possibly ways that adversaries can interact with interleaved protocol runs. The different kinds of analyses therefore tend to find different kinds of attacks resulting from different kinds of flaws.

Edited by David Basin 6 months ago

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

I see that, thanks. In fact I have read that in the responses here an in 1B, but I thought it was more a lack of the existing software than a principal difference.

I will thus replace the complete sentence with the statement of the clear advantages of the automatic approach with an explanation of the different kinds of flaws that the two approaches reveal. Probably tomorrow.

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

Would it make sense to add something a bit more general in the spirit of "security proofs are an ongoing process, and not a stamp obtained once and for all"? I think this may be participating to a general trend in the report that we are building.

In particular, apart from the challenge of proof correctness, proofs are established w.r.t. security definitions and models that are very much adhoc for the moment: there are many variations of them, and they often need to be adapted to the protocols that are analyzed. As a result, it is quite plausible that, at some point, someone realizes that a definition does not capture some important security aspects, and that a different proof might be needed.

Having a proof is definitely difficult an important, but it is also an important challenge to express and evaluate whether the properties that are proven are the ones that are needed.

David Basin @David.Basin · 6 months ago Developer

I support @Olivier.Pereira 's suggestion. There has been a lot of (excellent) work to systematize the community's knowledge of how verifiability and privacy should be defined in voting. But this is still an ongoing process.

Christian Folini @christian.folini changed title from **Discussion E - Formal Methods (Block 1 - Cryptographic Effectiveness)** to **Discussion 1E - Formal Methods (Block 1 - Cryptographic Effectiveness)** 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you David and Olivier. I've have reworded my summary based on your welcome feedback. I hope it no longer prioritizes machine checking over the manual approach but highlights the different characteristics in a balanced way. Feel free to comment or make suggestions.

David Basin @David.Basin · 6 months ago Developer

In the sentence: "Manual checking is also imperative since the two different kind of analyses - computational and manual" it would be better to contrast "computational and symbolic".

In "The computational analysis is strong with respect to the cryptographic functionality, the manual checks is more detailed with respect to the protocol's control flow and the possible ways that adversaries can interact with interleaved protocol runs." I would change "the manual checks" to "the automated symbolic analysis".

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you. Updated.

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

This looks very good to me!

Christian Folini @christian.folini mentioned in issue #18 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 1F - Breaking Cryptography (Block 1 - Cryptographic Effectiveness)

Reference to originating discussion block

Block 1 - Cryptographic Effectiveness, thesis points (4), (9)

Question

Assuming the real-world needs include protection from powerful organizations and state adversaries: how probable is it that such adversaries will rather find an attack based on the protocol specification (by finding a mistake or by breaking a cryptographic hardness-assumption) rather than the underlying specification, the code or the infrastructure? Assume reasonable key-lengths and no quantum computers. Relate your answer to the degree of scrutiny performed (e.g. number of reviewers). If you think it is anyway rather likely that attackers will find an attack based on the protocol specification, then why invest in cryptography at all?

Quantum Computers will be revisited at a later stage of the dialog.

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 06, 2021 7 months ago

Christian Folini @christian.folini added Block-1 Cryptography labels 7 months ago

David Basin @David.Basin · 7 months ago Developer

Take a look at SSL/TLS: we have seen numerous side-channel attacks such as variants of Bleichenbacher's attack (e.g., https://ieeexplore.ieee.org/abstract/document/8835216?casa_token=Biqk1_De_WUAAAAA:uoGXC7ZSzJu1bebgqrasQh-n0WQnrQEYtlR3Qvza3hL_hsNO5p0Asy88EigcOJ-FnX1uU3i-RHY) implementation attacks (like heartblead), design attacks like state-machine attacks (https://mitls.org/pages/attacks/SMACK). A Nation State Adversary will of course use whatever attacks are available of any kind. It is difficult to speak about likelihood here: the experience with TLS shows that all kinds of attacks arise in practice, and TLS is conceptually simpler than voting as it "only" establishes a secure channel.

Christian Folini @christian.folini assigned to @christian.folini 7 months ago

Christian Folini @christian.folini unassigned @christian.folini 7 months ago

Bogdan Warinschi @Bogdan.Warinschi · 7 months ago Developer

A powerful (e.g. state sponsored) adversary would use all of the attack surface available from specification to implementation, operations and platform. Ensuring the soundness of all of the aspects is equally important: a flawed implementation of a perfect specification or a perfect implementation of a flawed specification seem equally problematic. So I don't really understand the trade-off hinted at by the question.

Carsten Schuermann @Carsten.Schuermann · 7 months ago Developer

1. A state adversary will use any weakness of infrastructure or protocol to achieve its goals. Therefore if you ask how probable is it that the adversary will take advantage of a vulnerability present in the protocol, the answer most likely is 100%, in both cases. You ask how probable, the answer is equally probable.
2. Another concern is defending public confidence. If the protocol or the infrastructure are vulnerable, and it is known to an adversary, the adversary can use/misuse this information with the intent to cast doubt and destroy public confidence in the election result, for example, by alleging having broken through the perimeter.

Reto Koenig @Reto.Koenig · 7 months ago — Developer

Here we (e-voting group BFH) go with the statement of Carsten for question D, which states that cryptography shifts the attack vector towards the trust-assumptions. The better they are established and used, the more expensive an attack gets and thus renders it an economic question.

Edited by Reto Koenig 7 months ago

Collapse replies

Christian Folini @christian.folini · 7 months ago — Maintainer

For reference, here is a link to Carsten's response to question D.

Reto Koenig @Reto.Koenig · 7 months ago — Developer

Thanks for the advise... I just incorporated it in our text... will do so for future references.

Christian Folini @christian.folini · 7 months ago — Maintainer

No worries. Glad to help if I can.

Oliver Spycher @oliver.spycher · 7 months ago — Maintainer

Thank you for your answers! Given the statements here and in the other threads, I'd like to follow up with a new question.

How likely do you think it is that a powerful organization (the bad guys) is able to invalidate one of the important assumptions cryptography relies on without the research community (the good guys) knowing? (e.g. "hashes behave like random oracles", "discrete logarithms are impossible to find")

Similarly, how likely do you think it is that a powerful organization will know a serious flaw in a well scrutinized protocol specification long before the research community?

Collapse replies

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago — Developer

to answer your questions directly from my point of view

1. it must be considered and taken into account. But I would not rate the probability of occurrence too high
2. what do you mean with "long before"? Using heartbleed as an example: the vulnerability existed for more than two years until it was discovered by two independent researchers. We don't know if a "bad-guy" noticed it before and he probably wouldn't have communicated it to us because of the added value it gave him. I think in widely used protocols and enryption-mechanisms (primitive, etc.) the probability is rather small but present.

Oliver Spycher @oliver.spycher · 7 months ago — Maintainer

1. We can only "take it into account" by trying to think about the likelihood. (Replacing standard primitives and assumptions with something non-standard would make everything even worse.) If the likelihood seems high, then we would need to conclude that crypto per se does not offer protection against agencies. If the likelihood seems small, then we need to think whether it is small enough. I don't know how else we could take this into account.
2. I agree, we will never know if an agency was faster than the 2 researchers. But it seems that the agency knowing long before becomes increasingly unlikely with the number of insentivized people from the public that were looking for vulnerabilities over the two years. Would you agree? (What might the ratio of insentivized "good guys" vs "bad guys" have been at finding looking for OpenSSL vulnerabilities that ultimately led to finding heartbleed? Are we dealing with hundreds of agency-employees vs. few researchers and practioners from the public? Is a similar number of agency employees trying to find a mistake in protocol-specifications for a verifiable mix-net? What might the ratio be in that case?)

David Basin @David.Basin · 7 months ago — Developer

I think this is possible but it is more likely that the bad guys exploit implementation choices that have subtle consequences for the security of the primitives being used, so that they end up being less secure than theory would suggest. As an example, attacks are

possible by a nation state adversary when weak Diffie-Hellman parameters are chosen, see https://weakdh.org/imperfect-forward-secrecy.pdf. Now this attack was fortunately found by an academic group. If it was found by a nation state adversary, it is unlikely we would hear about it: it would just be exploited.

As for security protocol specifications: they are notoriously difficult to get right. Powerful bad guys have their own cryptographers and analysis tools. They are smart, well-funded, and dedicated. Moreover, they are focused and will analyze precisely what matters for their purposes (e.g., compromising elections in Switzerland). If they look hard enough, and long enough, they are likely to find issues, in particular given how protocols are developed today (NOT hand-in-hand with proofs).

Christian Folini @christian.folini · 7 months ago Maintainer

I'd like to call on all @experts to join the discussion here. @oliver.spycher 's follow up question makes this thread even more fundamental.

@David.Basin has already responded and I hope we can get more opinions on this question.

Vanessa Teague @Vanessa.Teague · 7 months ago Developer

I agree with everything that has been written above, particularly @David.Basin 's observation that it is possible that the bad guys find and exploit problems in the foundational cryptographic assumptions, but more likely that they find and exploit attacks specific to the protocol or implementation or other details specific to the individual case.

One reason for this is the different degrees of scrutiny that are applied to widely used primitives and assumptions vs particular protocols used by a particular country. One of the main reasons for using established primitives, modes and implementations for things such as encryption is that we assume such a thing has had significant peer review by numerous people in many countries, often for years. It's very hard to achieve a comparable level of examination for a protocol used by one country, even for something as important as e-voting.

So I would refine @David.Basin 's statement slightly and say *assuming that the primitives are well-established and standard* then the bad guys are more likely to exploit something else (such as implementation details, protocol weaknesses, etc).

Vanessa Teague @Vanessa.Teague · 7 months ago Developer

And I said 'primitives' but likewise for 'foundational cryptographic assumptions' as in @oliver.spycher 's question.

Florian Egloff @Florian.Egloff · 7 months ago Developer

@David.Basin said: "Powerful bad guys have their own cryptographers and analysis tools. They are smart, well-funded, and dedicated. Moreover, they are focused and will analyze precisely what matters for their purposes (e.g., compromising elections in Switzerland). If they look hard enough and long enough, they are likely to find issues, in particular given how protocols are developed today (NOT hand-in-hand with proofs)."

I agree entirely. Slightly orthogonal to the cryptography debate, but triggered by the example of focus: "e.g., compromising elections in Switzerland", I would add that, even for a smart, well-funded, dedicated attacker (likely a state's SIGINT agency) that goes after internet voting, there is some utility in focusing on building capabilities that will last (time) and that will be reusable across countries (scale). From an offensive standpoint, it thus makes sense to target a joint technology supplier. Picking up @Vanessa.Teague 's point: to the extent that building blocks of protocols are reused across countries, those would be the ones I would expect attackers to focus their efforts on.

Collapse replies

Oliver Spycher @oliver.spycher · 7 months ago Maintainer

You say "to the extent that building blocks of protocols are reused across countries, those would be the ones I would expect attackers to focus their efforts on."

Clearly, reused building-blocks are more interesting targets for finding exploits. But aren't these buildng-blocks interesting for the good-guys and the bad-guys equally? What do you think, would the added scrutiny due to reusing a building-block render a building-block rather more or rather less likely to be actually exploited? What would your answer depend on? Please also try to relate your answer to verifiable mix-nets, which are not standardized but yet important for internet voting. (@Florian.Egloff @Vanessa.Teague @David.Basin and everybody else!)

Florian Egloff @Florian.Egloff · 6 months ago Developer

Just to clarify: My response also included targeting joint "technology providers". So it does not necessarily have to be attacking the primitives, but rather, the actual implementation thereof in a specific technology provider. In e-voting, you have the offensive benefit of a small market, where few suppliers will exist (assumption on my part - I do not have a market study to back it up). If there are suppliers delivering technology to multiple countries of interest, that's where I would focus my efforts as an attacker. As for the

(in)equality of offensive/defensive interest, I think the answers below give a more qualified answer with regard to cryptography than I could attempt. I defer to them.

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

Thanks for clarifying!

Reto Koenig @Reto.Koenig · 7 months ago Developer

As an attacker, the 'sweet' spots to look for are definitely at the boundary where the attack-model is blocked by the trust-assumptions. This results in a simple equation: The stronger the trust-assumptions, the weaker the operational system in case of an attack.

So usually the 'good-guys' stop looking and hinder themselves at challenging the overall system by the 'blocker'/'spoiler': *'That is within the trust-assumptions'*. But it is the very exact location the 'bad-guys' start their work. This results in two distinct fields of research separate and reunite along the way from protocol level to operational system level.

Unfortunately though, the 'bad-guys' 'publish' their works in a different way than the 'good-guys' do, both having a major impact on operational level.

This way, it is a definite yes from our side! (:))

It is very likely that there are known serious ~~flaws~~ shortcomings in the well scrutinized protocol to find within trust assumptions that are too strong.

Here some examples probably stating very low-hanging fruits for NSAs, covering the 'correctness' of an election outcome at operational level:

Printing authority is fully trusted, GMP-library runs only in favour of the 'good-guys' and does not 'sing' for the 'bad-guys', CPU (Chipsets) does not spill/store secrets, memory content is not accessible via an 'alternative' channel, The power-supply does not spill/store secrets, the discrete-log is safe within *that* setting, ...

Oliver Spycher @oliver.spycher · 7 months ago Maintainer

So widely used primitives and assumptions are safer because of the vast scrutiny they have been undertaken. Let's look at zero-knowledge proofs. They have been widely researched but not scrutinized to the same degree. I wonder if we can think about the likelihood of agencies being the first to find a weakness and keeping it secret. Lets say we have a team A of 10 and a team B of 100 incentivized participants, both challenged to find a flaw in some protocol. Clearly team B has better chances to be first (10x greater?), but team A could also make it. How would you estimate the proportion of good guys over bad ones in the real-world? We know that the good guys travel around the world, go to conferences, speak openly within a broad community. Their incentives might be professional success as researchers, public recognition, curiosity, sense of responsibility, maybe also bounties. The bad guys are payed by someone who can benefit from breaking zkp but they have to remain in shadow and are not to advertise their finding if the people paying them want to use it. How much better will they do than the good guys?

Collapse replies

Tobias Ellenberger @Tobias.Ellenberger · 7 months ago Developer

Let's speculate. Mathematically, the probability in the example is higher in advantage of the bad-guys. Assuming that the bad-guys can find a weakness and keep it secret, the following possible questions arise:

- for how long will the vulnerability remain undiscovered?
- what is the first thing to be attacked with the new possibilities (most lucrative goal)
- how to make the most profit from the weak point (after all, "there" only counts the money)

If a vulnerability is found in a zero-knowledge-proof, it could be used to do lucrative things to get money directly. For this purpose, as many different targets as possible would be attacked simultaneously, which will hopefully be noticed by the good-guys within a shorter time (estimate?). As a primary goal I do not think of evoting. to the last question: don't think that the bad-guys do it better than anyone else, but they may have a different (better?) motivation. In the direct context of e-voting I do not (yet) see the ROI for the potentially big effort.

Bryan Ford @Bryan.Ford · 7 months ago Developer

The profit-motivated "bad-guys" seem to have at least a couple natural advantages in the example above: (a) greater resources

(including strength in numbers) due to the funding/incentives available, and (b) the availability of the strategic choice to keep the vulnerability secret without worrying about small matters like ethics.

Potential imbalance (a) is important, but can potentially be counterbalanced in several ways. Going back to the topic of standardization and breadth of audience and scrutiny for components: for crypto like AES and protocols like TLS that a ton of companies around the world depend on all the time, there is at least a lot of potential incentive and funding on the good-guys side too to scrutinize such components and keep (or make) them secure. If E-voting systems helped develop and use standards for zero-knowledge proofs, verifiable shuffles, bulletin boards, etc. that were sufficiently general and broadly-designed to be of interest to a lot of non-e-voting applications as well as e-voting, then the e-voting community would get similar benefits in strength of interest in and potential financial backing for the good-guys as in TLS. Generality and audience breadth of components helps create beneficial network effects in security.

This "potential" financial incentivization on the good-guys side doesn't necessarily or always get translated into real financial investment in securing protocols or implementations, unfortunately. OpenSSL and Heartbleed is the classic example of huge failure in that regard, in that everyone was using and depending on the same library and just hoping/assuming it was secure but no one was actually funding it. That's somewhat fixed now for OpenSSL but in general there needs to be workable funding models to ensure that "potential funds" for good-guys discovering and fixing bugs in open standards and components gets turned into actual financing for security.

Potential imbalance (b) above, namely the bad-guys' ability to use and profit from secrecy as a tool, is also important. One potential way to counterbalance this in favor of the good-guys, of course, is a suitable and sufficiently well-funded bug bounty. And as I discussed in some length in my answer to question 4.7, to skip ahead a bit, appropriate use of multi-implementation diversity integrated with bug bounties in the right way can potentially turn the tables around in favor of the good-guys with another exponential-over-linear advantage. This is because if the bug bounty program rewards anyone who finds and responsibly discloses a weakness they find in *any one* component, they get a certain immediate reward. But a profit-motivated hacker who finds a vulnerability in (say) one control component and wants to exploit it secretly must first wait and find corresponding or jointly-exploitable vulnerabilities in the other three control components before being able to do so. The exploitation-motivated hacker, upon accumulating the first one or two secret vulnerabilities, risks losing them at any time to some other hacker who just wants the bug bounty and reports the bug immediately, and is effectively playing a "gambling game" that the hopefully rare event of finding an exploit in one control component pays off *four times* in a sufficiently short time period. Thus, the hacker or organization who wants to exploit the vulnerabilities had better think the exploit is *really* valuable if they're going to forgo the quick-and-certain bug bounty profit and instead wait for the slow-and-uncertain prospect of collecting the whole set before someone else finds discloses the vulnerabilities. This is essentially the idea that the Hydra framework applies to smart contracts, but it's powerful and potentially applicable here as well.

David Basin @David.Basin · 7 months ago — Developer

@oliver.spycher 's question is difficult for me, at any rate, to give a definitive answer to. It is difficult to get good historical data on how successful bad guys are (who, depending on your perspective, might be good guys, e.g., your security agencies, signals intelligence, ...). It is mostly anecdotal. Every so often we get a treasure trove like the Snowden revelations where we see what powerful bad guys are really capable of: tapping all transport links, traffic diversion, storage and large-scale analysis, backdooring hardware, software, and libraries, coercion (legal or otherwise) of major hardware/software/service manufactures to work with them, etc. Good guys don't have these options.

Maybe where the playing field is most level is with basic primitives (e.g., ZKPs) as mathematical objects and the study of their mathematical properties, as opposed to their realization in libraries and systems. Least we forget, the implementations of very basic cryptographic functionalities, even in Switzerland, may be under the control of intelligence agencies.

Philippe Oechslin @philippe.oechslin · 7 months ago — Maintainer

There is an interesting point to make regarding Snowden's treasure trove mentioned by @David.Basin . There is no indication of the NSA breaking any crypto primitives.

They did have resources and motivation but they went for implementation bugs or abusing their position to do eavesdropping, forcing collaboration or weakening crypto standards.

My conjecture is that the bad guys are not better at breaking crypto primitives than the good guys. Given enough time, the good guys should be able to find the same bugs than the bad guys.

Carsten Schuermann @Carsten.Schuermann · 7 months ago — Developer

Yes, but there is some suspicion that powerful Nation State actors do all they can to stay ahead. See Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

David Basin @David.Basin · 7 months ago — Developer

I agree with @phiippe.oechslin here about Snowdon: the NSA found easier ways to break into systems.

There are other examples though that have made their way into the public domain about breakthroughs in cryptography by intelligence agencies that stayed secret for many years. E.g., the independent development of RSA by Clifford Cocks at GCHQ in England. This remained a secret for ca. 24 years. This is relevant for other questions too, e.g., does the NSA have quantum

supremacy with respect to factoring integers? How should we know?

Philippe Oechslin @philippe.oechslin · 7 months ago Maintainer

@Carsten.Schuermann, in the particular case of breaking 1024 bit Diffie Hellman, the NSA might have been able to do it because of the massive computing power and specialized hardware they could have had at their disposal. But still, as described in the paper (sec 4.2) and the leaked NSA documents, the attack would only work if they were able to get the IPSec PSK through eavesdropping or from databases of known PSKs. Also, there is no evidence that they actually did discover the discrete logarithm attack.

So, again, it does not seem that they were smarter, but they had more resources and motivations.

Christian Folini @christian.folini mentioned in issue #14 (closed) 7 months ago

Srdjan Capkun @Srdjan.Capkun · 6 months ago Developer

My concern is not so much in breaking the foundations of these ZK primitives (e.g., invalidating their assumptions). It is related to the overall lack of the time spent, expertise available to understand, correctly implement and maintain such (e.g., ZK) systems. Using the Swiss voting as a guinea pig for the deployment of ZK on a large scale seems risky. Going back to the original question of this thread, "why invest in cryptography at all?", we do have deployments of security technologies that are much more mature and where vulnerabilities if found are less likely to be so severe.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

@Srdjan.Capkun: You paint a very negative picture of the state of the research of ZKP. Do you see alternatives to their use here, or is it just that you think ZKP are not researched enough to use them for important systems as this?

I'm asking since ZKPs have been researched a lot and there are uses beyond voting like crypto currencies for example. I am not putting their importance on the same level as the foundations of our democracy, but given the high monetary values in several of these currencies attacking their cryptographic foundations would yield very lucrative returns.

Also, I am not quite sure I understand your closing sentence correctly. Here is what I read: ZK systems are still experimental and they should first be tested on systems / uses where a vulnerability would have less severe consequences. Correct?

Bryan Ford @Bryan.Ford · 6 months ago Developer

I read Srdjan's concern as not so much with the state of ZK systems *research* as with practical ZK systems engineering and maintenance in deployed systems, and in that I agree with him. The theory seems quite solid as far as it goes, but any ZK proof beyond a digital signature is seen as pretty exotic, usually not present at all in the most standard and well-maintained crypto libraries, usually developed and maintained in application-specific contexts with moderately narrow "niche" audiences, etc.

But I also agree with you Christian that this is only the current situation and there are signs that may rapidly change for the better. For one thing, the blockchain/cryptocurrency bandwagon, as noisy and scam-ridden as it is, has directed a lot of broader awareness, interest, and funding toward more general ZK systems including verifiable shuffles. (See for example CoinShuffle and the many variant schemes and cryptocurrencies such as Monero that build on shuffles.)

General ZK proofs, and shuffles especially, don't *need* to be exotic things that only cryptographers and E-voting people have ever heard about and that nobody puts into libraries or standards. There are many applications outside of E-voting that naturally can and want these primitives, and some are increasingly adopting them. It's partly just a matter of time. But also, as I've discussed elsewhere, industry- and state-level participation in and assistance with standardization efforts (like what the IRTF's CFRG is doing for other crypto primitives) could potentially speed that up a lot too. Suppose, for example, that representatives of one or two governments interested in E-voting were to tell CFRG that their governments would mandate the use of standardized ZK systems if CFRG (say) were to produce suitable standards. I bet the privacy-preserving cryptocurrency people and other industry players would jump into that as well, and then those ZK standards would start appearing in a lot more well-maintained and widely-scrutinized crypto libraries.

Christian Folini @christian.folini changed title from **Discussion F - Breaking Cryptography (Block 1 - Cryptographic Effectiveness)** to **Discussion 1F - Breaking Cryptography (Block 1 - Cryptographic Effectiveness)** 6 months ago

Srdjan Capkun @Srdjan.Capkun · 6 months ago Developer

Bryan, thank you for clarifying. I believe that we need to use mature technologies, not only in research sense, but also in terms of deployment and available talent. Maybe e-voting should not be a test case for 'exotic' (be it research wise or deployment wise) technologies.

Christian Folini @christian.folini · 5 months ago Maintainer

We let this discussion settle for a few weeks to see how the other discussion blocks would develop. This has happened with a clear statement for more implementation diversity in block 2 and an extensive exploration of the pros and cons of public bulletin boards.

But it is important that we return to this crucial question in time, so we can actually write up a summary.

Here is my draft:

Assuming the real-world needs include protection from powerful organizations and state adversaries: how probable is it that such adversaries will rather find an attack based on the protocol specification (by finding a mistake or by breaking a cryptographic hardness-assumption) rather than the underlying specification, the code or the infrastructure? Assume reasonable key-lengths and no quantum computers. Relate your answer to the degree of scrutiny performed (e.g. number of reviewers). If you think it is anyway rather likely that attackers will find an attack based on the protocol specification, then why invest in cryptography at all?

Powerful organisations and state adversaries with sophisticated technical knowledge and resources have the realistic capability to find and secretly exploit the most subtle flaws in an electronic voting system successfully.

Well established and standardized cryptographic building blocks that have seen a lot of scrutiny are more likely to resists these attacks. It is therefore beneficial to use mature technologies, including cryptographic primitives, that have been standardized, are widely deployed, and are supported by talent. Absent such maturity, it would be wise to support the creation of standardised cryptographic building blocks and well-maintained crypto libraries supporting internet voting.

Attackers are more likely to attack non-standardized aspects of the protocol, the implementation of the voting system or the infrastructure used to run the system. This is because these elements have seen less public scrutiny. The trust-assumptions of the system are also natural points for attacks, namely when they are too strong.

Another characteristic of the attackers in question here is their ability to keep weaknesses for themselves. It is therefore useful to incentify research that helps to reveal and fix weaknesses in the building blocks, but also the specification, the implementation and infrastructure used to run the system.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT] Typo ("resists" -> "resist")

[EDIT] Updated the first paragraph with a new proposal by @Bryan.Ford and extended the 2nd paragraph after detailed feedback by @Florian.Egloff.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

1§: I suggest reformulation to the following, for language nit-picky reasons: Powerful organisations and state adversaries have the knowledge and resources necessary to attack an electronic voting system successfully.

2§:

- change "resists" to "resist".
- I would suggest adding some language that follows @Srdjan.Capkun 's differentiation in (#7 (comment 585)) about maturity. Perhaps one could reformulate to (this is just a suggestion basedon @Bryan.Ford and @Srdjan.Capkun 's comments above, not an "expert comment": Well established and standardized cryptographic building blocks that have seen a lot of scrutiny are more likely to resists these attacks. It is therefore beneficial to use mature technologies, including cryptographic primitives, that have been standardized, are widely deployed, and are supported by talent. Absent such maturity, it would be wise to support the creation of standardised cryptographic building blocks and well-maintained crypto libraries supporting internet voting.

the rest seems ok to me.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Florian. I have fixed the typo.

With the other two changes, I see a slight change of weight and I am not sure everybody is following that. Let's see if we get more comments on that. Maybe from the other participants in this thread.

Reto Koenig @Reto.Koenig · 5 months ago Developer

> 1§: I suggest reformulation to the following, for language nit-picky reasons: Powerful organisations and state adversaries have the knowledge and resources necessary to attack an electronic voting system successfully.

We believe, that this exact 'Belief' of 1§ is a successful attack on the electronic voting system. This passive attack that results in not building the system at all, or loosing faith in it.

It really depends on how the system is built, in order to know what and where to attack and the meaning of 'successfully'. If successfully is meant to change the outcome of an election unbeknownst to the rest of the world (or at least the democracy involved), that is a difficult task. Not theoretically impossible, but 'believable' impossible.

If we imagine building the system providing theoretical unconditional everlasting privacy and the systems processes are fully verified and dispute free, using trusted hardware (no, we still do not accept smartphones ;-)), then there is no attack vector available after the voter voted. To successfully attack the system, one has to attack prior to the voting phase. That is equivalent to a blind shot in Switzerland. If the adversary needs to be able to launch the attack dynamically (targeted), then the adversary needs to be able to control almost all organizational means... human means. And that is at least as complex as attacking any other voting channel.

If the cryptographic building blocks are used in a sensible way, they all rely on organizational means. So no adversarial lock picking but social engineering would be the attack-vector (=trust assumptions). If Switzerland has been able to cope with that so far against all adversaries (?has it really?), then it can cope with it for any other well designed voting-channel.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I see no disagreement for §2.

For §1 I am happy to leave it as is. I personally just don't like the passive grammatical construction, I.e. who is "seeing"?

Edited by Florian Egloff 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree with your concerns @Florian.Egloff about how that sentence is written. How about replacing it with:

> Powerful organisations and state adversaries with sophisticated technical knowledge and resources have the realistic capability to find and secretly exploit the most subtle flaws in an electronic voting system successfully.

This phrasing (a) eliminates the passive voice, and (b) weakens the claim slightly from implying that they can break the voting system "no matter what" to stating that they can break the most subtly-flawed voting systems. (Which may be almost the same thing in reality since no system is perfect, but I do think the semantic difference is important and agree with @Reto.Koenig that the absolute claim makes it sound like we should just give up now.)

Christian Folini @christian.folini · 5 months ago Maintainer

I think this is a well balanced proposal @Bryan.Ford. It takes up the reasonable concerns that @Florian.Egloff put forward and it does not change the weight too much.

I'm giving @Reto.Koenig some time to respond before I will adopt this.

@Florian.Egloff and §2: true that no negative feedback to the idea to clarify this.

How about this replacement?

> Well established and standardized cryptographic building blocks that have seen a lot of scrutiny are more likely to resists these attacks. It is therefore beneficial to use mature technologies, including cryptographic primitives, that have been standardized, are widely deployed, and are supported by talent. Absent such maturity, it would be wise to support the creation of standardised cryptographic building blocks and well-maintained crypto libraries supporting internet voting.

I know you on holiday @Florian.Egloff. So I am leaving this open for a few more days. If I am not hearing from you or anybody else on this question, I am going to adopt it in the summary.

Edited by Christian Folini 5 months ago

Reto Koenig @Reto.Koenig · 5 months ago Developer

We do not want to spoil any of the replacements, in fact we are in line with the latest replacements proposed here.

Our main point is, that it does not only boil down to the sheer 'fire-power' of an adversary, that leads to a successful attack but also the time-window and organization available as well as the perfect timing. This allows a system to remain secure even if the used cryptographic components are practically vulnerable, as long as it denies the adversary one of the other aspects just mentioned. In our perspective, a system is much harder to be attacked successfully, if it is designed in a way that its security goals must be protected by cryptographic means during the online phase (say 20 days) only instead of 'forever'.

Concluding, we do agree that we must cope with an adversary having the realistic capability to find and secretly exploit the most subtle flaws in an electronic voting system successfully, but there is more required than 'just' that in order to result in a successful attack and any system design and operational procedures in place should take that into account too.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for chiming in @Reto.Koenig and thank you for the support for the 2 replacements. I have thus incorporated them into the text.

Your explanations make a lot of sense, they have been upvoted and I see how they add a new aspect to the conversation here. I read it as system design and operation procedures can be used to block an attack even if the crypto is broken.

Yet I am struggling on how to integrate this into the summary very late in the process as it looks like a new aspect to me. We are facing several challenges:

- When I read the question again, then it focuses on the cryptography and the attacks, not the exploits.
- If we do not find a very good wording, then the updated summary with this line of thought could be read as "experts are trying to block state sponsored actors with a WAF instead of strong cryptography".
- It is very late in the process and thus a bit late for new arguments.

With all this being said, I would rather leave the summary as is and let it focus on *breaking cryptography*.

If you really think it has to be included, then we might have to an exercise and write to the 25 other participants in this conversation to vet a possible rewording in time.

Reto Koenig @Reto.Koenig · 5 months ago Developer

We agree here: it is not about *breaking the system* in place, but 'only' about *breaking cryptography* used within the system in place.

So, we will keep our more holistic view ready for another dispute... and thanks to the thumbs up, we know... we are not alone! ;-)

Edited by Reto Koenig 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Reto.Koenig. Much appreciated.

With this being out of the way and no additional pending feedback, it's time to wrap it up. I am closing this question.

I thank you very much for your constructive participation in this thread. It has been very intense, namely in the beginning, and we were not sure if we could ever come to a good conclusion on this front. I am glad it worked out in the end.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Update 2-diversity.md
Christian Folini authored 7 months ago

4e71d0d1

2-diversity.md 26 KB

Discussion Block 2 - Diversity to support security and trust-building

1. Introduction

Verifiability in Switzerland aims at detecting cases of large-scale fraud. Many computations cannot be verified by the voter but by the administrations or elected commissions. Their verification capabilities hinge on whether critical components they or their partners run are functioning correctly. Due to the trust model defined in Art. 5 in conjecture with chapters 4.1 and 4.3 of the annex, it must be possible to justifiably assume that at least one control-component, at least one verifier and the printing office are functioning correctly. These critical components are introduced in the desciption below.

2. Individual Verifiability

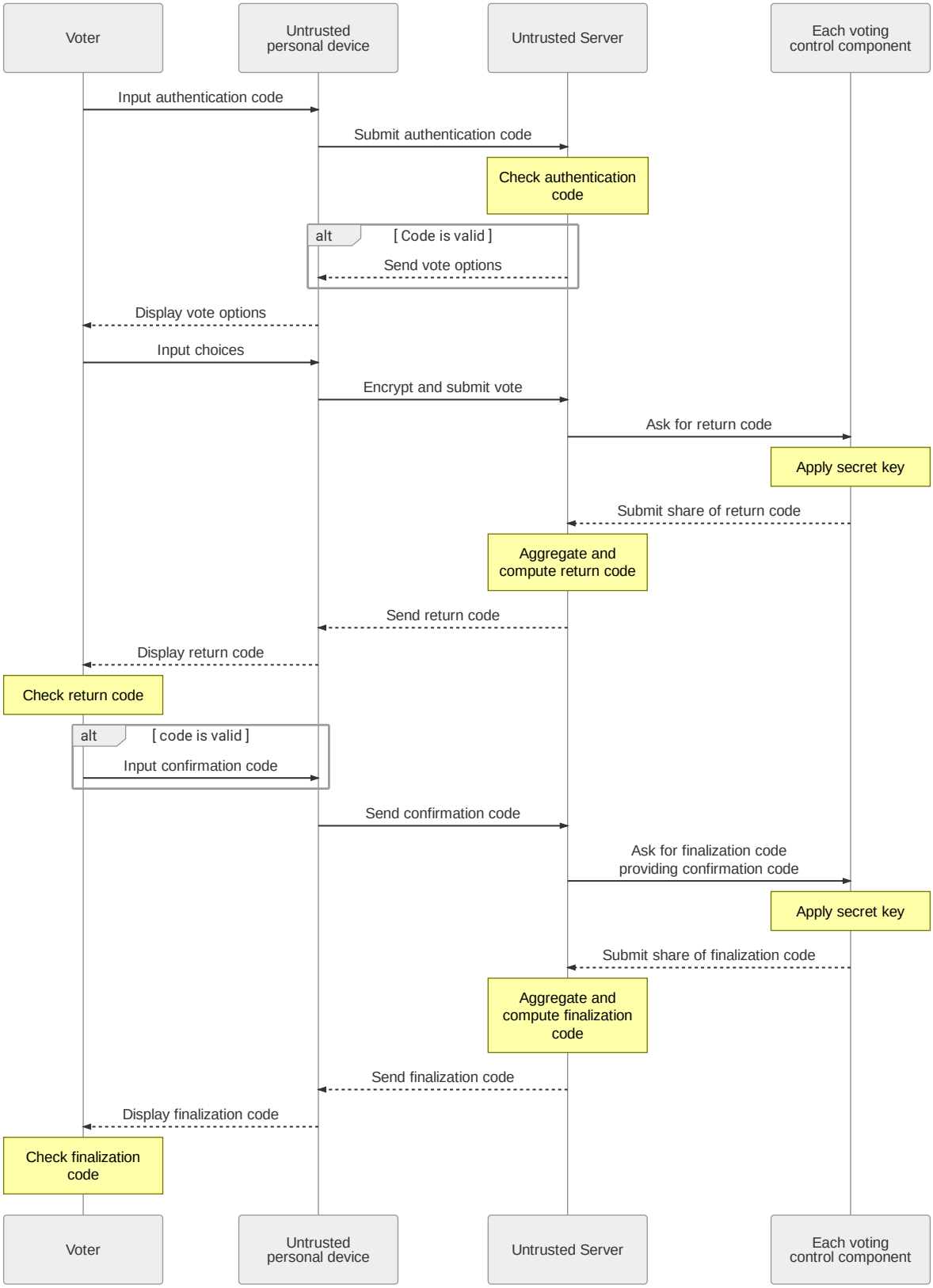


Figure 1 shows how “individual verifiability” is intended to work today. Any system compliant with the minimal requirements of the VEleS would work similarly.

The voters open their voting material that allows them to vote at the polling station, by postal mail or through the internet. The voting material contains instructions for each voting channel. For internet voting, the voting card contains the codes that are needed to authenticate and to verify that votes have been cast-as-intended due to “individual verifiability” as defined in Art. 5 VEleS in conjecture with Art. 4 VEleS and Chapters 4.1, 4.2, 4.3 and 4.4 of the annex.

After entering their authentication code, the voters fill in their ballot by clicking on answers (yes/no/empty) in popular votes, or on lists and candidates in elections. After confirming, the vote is encrypted and sent to the voting server. Each of the four control-components apply their distinct secret keys on the received data. They send back messages that allow the untrusted server to generate the return-codes related to the individual selections made by the voter. Due to the regulation, the content of the votes needs to remain secret, in particular the votes may not be decrypted. The return-codes are then displayed to the voters on their device. Voters are instructed to compare the displayed return-codes (distinct codes per answer and voter) with the ones received with the voting material and to report to the administration if the codes are not displayed correctly. The voters can then either change

their mind and vote by postal mail or at the polling-station, or confirm the internet vote by entering their confirmation code. Similarly as above, the control-components participate at generating the finalization code which is displayed to the voter for verification against the voting material. If the finalization code is not displayed correctly, voters can enter their confirmation code repeatedly until the finalization code is shown, or contact the administration.

Codes being displayed correctly holds the meaning that the vote will either be counted according the voter's intension or, due to universal verifiability (introduced in section 4), the administration or an elected commission will detect fraud and start an investigation.

3. Critical Components: Control Components for individual verifiability

One out of any of the four control components involved in generating the return codes and the finalization code must function correctly, i.e. at least one needs to fulfill the relevant tasks as defined by the cryptographic protocol specification. No software errors or tampering may be suspected to cause a malfunction. If the assumption that at least one of the control components performs according to the protocol is not believed to hold, the return codes do not have any meaning. The following list contains examples of the relevant functions underlying the effectiveness of individual verifiability:

- They must keep the keys they use to establish the return codes secret. An attacker who knows these keys could manipulate votes and display to the voters the codes they expect to see.
- They may only apply their keys once per voter, otherwise attackers could learn all the codes, manipulate the votes and deceive the voters.
- They must make sure that the information received by the voters complies with the protocol. For example, if the return-codes are already pre-computed on the user platform, then the control-components need to make sure that the pre-computation relates to the actual vote (this is done by checking a zero-knowledge proof delivered by the voting client). Otherwise, attackers could manipulate the vote but the precomputations would lead to the control-components establishing the return-codes the voter expects to see.

4. Universal verifiability

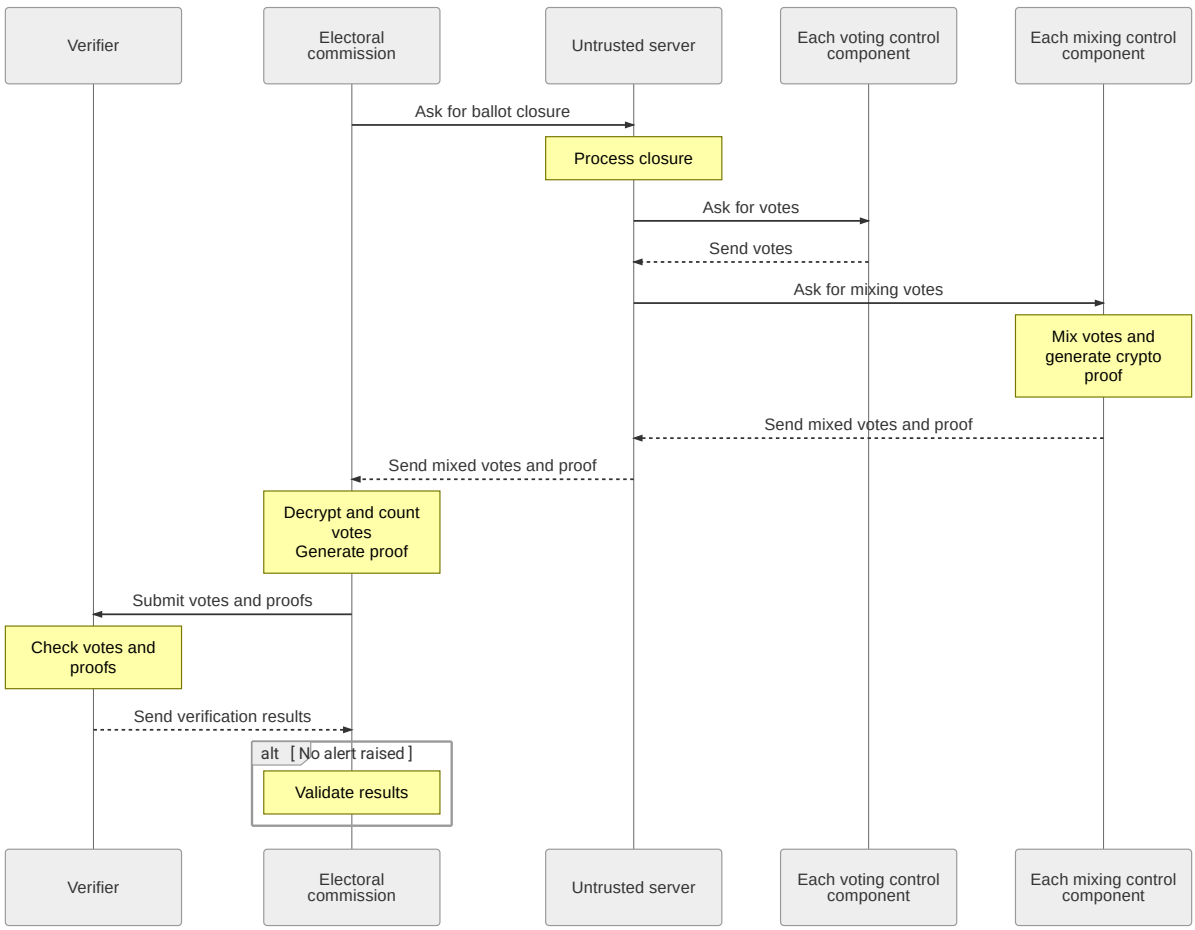


Figure 2 shows how "universal verifiability" is intended to work today (individual plus universal verifiability equals "complete verifiability", as set out in the VEleS and the annex).

After the voting period is over, a set of at least four control-components sequentially mix all votes and change (re-randomize) their encryption. They provide a zero-knowledge proof that no votes have been changed despite changing their encryption. After mixing, the votes are decrypted and the correct decryption is proved using zero-knowledge proofs. The secrecy of the vote is granted, assuming that one of the control-components will not reveal the permutation used at mixing.

Based on the data received from the control-components active during the voting phase, the data received from the control-components performing the mixing as well as the data received from the component used for the final decryption, the administration or an elected commission use the verifier to verify that all votes have been considered in the final tally correctly. To that end, the verifier checks the consistency of the lists of votes received from the control-components (thus it verifies that the votes have been recorded-as-cast) and it verifies the proofs of correct mixing and decryption (thus it verifies that the votes have been tallied-as-recorded). Typically, the verifier is a device that would be run in the premises of a canton.

5. Critical Components: Control Components for universal verifiability

In order for the administration or an elected commission to detect manipulations, one out of any of the four control components must work correctly. The following list contains examples of functions or tasks the control components have to perform correctly, i.e. according to the cryptographic protocol:

- The control-components that generate the return-codes must keep all votes or a digest until the end of the vote. If at least the correctly functioning control-component keeps all votes, then the correctly functioning verifier will expose an inconsistency in the votes, in the case where other control components change or delete votes. However, if all control components delete the votes or change them in the same manner, this will not be detected by the verifier, i.e. there might be no detection based on well-analyzed cryptographic means.
- The control-components that perform the mixing may not reveal the applied permutation. (This as well as the next point is to protect the secrecy of the vote, the verifier being able to detect manipulation does not hinge on these constraints.)
- Neither may they reveal the key's secrets that they use for changing the encryption. (This is to protect the secrecy of the vote, the verifier being able to detect manipulation does not hinge on this.)

6. Critical Components: Verifier for universal verifiability

In order for the administration or an elected commission to detect manipulations, one verifier needs to work correctly. The following function has to be executed correctly, i.e. according to the cryptographic protocol:

- Verifiers need to raise an alarm if the lists of votes differ across the control components or if a proof of correct mixing or decryption does not hold.

7. Critical Components: Printing Office for individual and universal verifiability

The printing office is needed to produce the voting material, including the codes for authenticating and for verifying that their vote has been cast-as-intended (“individual verifiability”). The effectiveness of individual verifiability hinges on the return-codes, the confirmation code and the finalization code being secret and impossible to predict. In practice, the tasks of the Printing Office are divided. The generation of the codes and other parameters are performed in the cantonal premises. The physical printing is done by a printing company (see section 8.3).

The printing-office holds the codes in plain-text. At the latest when the codes are being printed, there can be no cryptographic means to protect these codes from being divulged. The codes have to be protected by organizational means.

The VEleS does not make a clear cut as to whether the printing office is allowed to generate parameters itself (including the codes) and whether it is sufficient to rely on organizational means to ensure the unpredictability of such parameters. In case the printing office is used to generate parameters, then the effectiveness of individual and possibly even universal verifiability additionally hinges on the quality of the printing office’s random-number generator, its aggregating the inputs from the random-number generator correctly and printing the values accordingly.

In order for voters, the administration or an elected commission to detect manipulations the printing office needs to work correctly. The following list contains examples of functions or tasks the printing office has to perform correctly, i.e. according to the cryptographic protocol:

- Secret values, in particular the codes for verifiability and for authentication, must remain secret.
- Codes and if applicable other parameters must be generated correctly, i.e. according to the protocol specification.
- The generated values need to be further processed according to their generation, i.e. the printer must print these values correctly.

8. Critical Components in practice

The correct functioning of at least one control component and at least one verifier can be supported by:

- Scrutinizing the individual components
- Ensuring diversity among the components (different software and hardware)
- Increasing their number
- Keeping functionality simple and easy to analyze
- Appointing distinct groups of employees to operate each of the components
- Operating components in different premises

The correct functioning of the printing office can be supported by:

- Requiring the consent of at least 2 persons to obtain access that is further unnoticed
- Using offline machines when working with confidential information and destroying the data afterwards
- Scrutinizing the software and the processes

These points are addressed in the VEleS annex to a certain extent but not necessarily as a strict requirement.

In order to have something concrete to relate to, we summarize the features and modes of operation of the critical components as planned by the cantons when aiming for offering internet voting using system of Swiss Post. We point out that the features outlined here do not stand in conflict with the provisions of the VEleS and the annex in their current state.

8.1 Control components in practice

Control components as planned to be used during the voting phase:

- Implemented as physical servers (no virtualization)
- Segregated access rights
- Segregated networks
- Different operating systems and CPU
- Each to be operated in the premises of the provider
- Each to run the same application-level software from the voting provider
- Software to be scrutinized and published

Control components used for mixing the votes:

- The same as above holds for three of these control-components
- The same holds for the fourth control-component, except that it was meant to be operated offline in the premises of the canton

8.2 Verifier in practice

- One verifier was to be used
- Operated offline in the premises of the canton
- Due to the tight schedule, software from the same provider as the voting system was to be used. The software was to be scrutinized and published.

8.3 Printing office in practice

- The planned functionality attributed to the printing office can be distinguished as follows:
  - Parameter generation was to be done in the cantonal premises; software from the voting provider was to be used.
  - The actions of the actual printing office was reduced to decrypting the data received for printing, the printing itself and enveloping the voting cards. The cantons were to mandate their own printing service (the voting provider was not meant to print the voting cards).
- Some parameters underlying the effectiveness of verifiability were to be generated in a distributed way, but not all. For instance the return codes as printed on the voting cards were to be selected based on random generator used by the component run in the cantonal premises.
- Parameter generation was not to be verified
- Parameter generation and printing was to be performed under the surveillance of at least two persons (four eyes principle)
- Private data was to be held on machines that were disconnected from any network (the computers and the printing machine were planned to be offline)

9. Related Questions

The related questions are labelled Block-2.

9.1. Individual links to related questions

- Block 2 Discussion A - Independent Software for Control Components used for return-code generation
- Block 2 Discussion B - Independent Operating Systems for Control Components used for return-code generation
- Block 2 Discussion C - Independent CPU for Control Components used for return-code generation
- Block 2 Discussion D - Control Components used for mixing and verifier

The Printing Office is not really covered in this discussion block. However, the 3rd discussion block will be dedicated to this topic.

10. Questionnaire

The thesis is based on the questions 2.1.4, 2.1.5 and 2.1.6 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.1.4	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.1.5	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.1.6	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

11. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 2A - Independent Software for Control Components used for return-code generation (Block 2 - Diversity)

Reference to originating discussion block

Block 2 - Diversity

Question

In their answers to question 2.1.4 of the questionnaire, the invited experts acknowledge the added value in running software from independent providers. However opinions differ with regard to whether the risks emerging from the added complexity would overrule the added value with regard to verifiability. We would like to dig deeper in order to find out whether efforts should be put into introducing software from a different provider for at least one of the four control components used during the online phase (see section 3 and the first point in section 5).

Imagine Bob is an employee of a cantonal administration. It is his job to convince himself and others that internet voting is sufficiently secure. The control-components used in the online phase are currently meant to run the same software. The software has been scrutinized to some degree and is currently being refactored. Auditing of the refactored software by independent engineers and cryptographers is planned and paid for, as well as publication of source code and documentation in order to allow for further scrutiny.

Bob really needs to get it right and Alice grants him additional funds. He thought about it and is now faced with a dilemma. One option would be to procure a new software for at least one of the control-components and to have it scrutinized. Another option would be to invest in further examination or other risk-limiting measures of the existing software. He is not sure about the added complexity of the first idea. Although the software would be operated by the existing voting provider, still, the provider of the new software would constitute an additional actor and the new software would need to function well along with the existing one. On the other hand, diversity is a best practice in internet voting.

Would you advise Bob to procure the additional software or should he invest the money in other ways of limiting the risks related to the existing software?

Assume the underlying crypto-protocol is specified on a reasonable abstraction level and that a technical interface specification (data formats, timing constraints) is available.

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 01, 2021 7 months ago

Christian Folini @christian.folini added Block-2 Cryptography labels 7 months ago

Christian Folini @christian.folini changed due date to February 01, 2021 7 months ago

Bryan Ford @Bryan.Ford · 7 months ago Developer

As @Eric.Dubuis @Rolf.Haenni @Reto.Koenig @Philipp.Locher pointed out in their answer, we have well-understood and reasonably simple ways to reason mathematically about the benefits of diversity. (My PhD student Ennan Zhai's independence-as-a-service work a few years ago developed this further toward more automated infrastructure risk analysis.)

Just to reiterate here for convenience, assuming n control components run truly-independent software and p is the probability of a failure (e.g., exploitable bug) in one of them, then the probability that all four fail is p^n. This is of course simplistic and the real situation is never that good due to limitations on independence and common-mode failures, etc. - but the important thing is that to the extent there is true independence and diversity, it creates an exponential cost-benefit curve. You invest only linearly more resources to get exponential reward in terms of decreased failure probability.

For example, just to illustrate, suppose for the sake of argument there is a 10% chance (p=0.1) of anyone (publicly or secretly) discovering a critical software flaw in one control component, which costs c CHF to develop. Developing four fully-independent control components increases the total cost to 4c, while in principle reducing the probability of a critical software flaw compromising the whole system to p^4 = 0.0001.

Now, there may be reasonable arguments that investing greater resources into a single software implementation to harden it further might be more beneficial. However, that depends on what the cost-reward functions of those other potential hardening investments are, and these cost-benefit reward functions might not be particularly easy to calculate, estimate, or measure empirically. Do the alternative hardening approaches provide exponential, or sub-exponential, or super-exponential rewards with further investment?

To build on the concrete example above, suppose for the sake of argument that investing 10x more CHF in analyzing and hardening a particular control component software implementation, using some suite of analysis and hardening techniques, decreases the probability of a compromise being discovered in that code by a commensurate factor of 10. That is, for a cost of 10c CHF, we have reduced the failure probability to p/10, i.e., 1% failure probability. Then by investing the full 10c budget into one control component, we get the overall system's failure cost down to 1%.

But suppose we instead developing four independent control components each hardened 1/4 as much, for a cost of 2.5c CHF per control component, keeping the same 10c CHF total cost. Hardening has reduced the failure probability of each control component by a factor of 2.5 to 4%. But diversity then reduces the overall system failure probability further to .04^4 = 2.56x10^-6. Thus, in this hypothetical scenario diversity wins over pure hardening of one software implementation by almost five orders of magnitude.

Of course this presumes the assumption above that the hypothetical (as-yet-unspecified) basket of hardening investments may reduce the probability of one component's failure only linearly. For example, it may well be that investing 10x the cost to develop a fully-verified software implementation with a mechanically-checkable proof will reduce the probability of an undiscovered flaw exponentially or even better, and not just linearly. Naively, in fact, a mechanically-checked implementation would seem to reduce the probability of an undiscovered software flaw to zero - but this is only if we neglect to consider flaws where the model or specification is flawed or fails to match reality, or bugs in the proof-checker itself, neither of which are at all unheard of. (Even though mechanical verification is not my area, in the early 2000s I managed to find a bug in Isabelle that let me prove 1=0 - that was fun.)

So to the cost-benefit tradeoffs of diversity against other hardening approaches, we ultimately need some form of evidence - analytical and/or empirical - that the other proposed hardening approaches reduce failure probability at least exponentially with respect to investment, if not better. I would love to be pointed to quantifiable evidence of this form for any given proposed basket of hardening techniques.

Note that I'm definitely not picking on mechanical verification here in particular, which I strongly believe is valuable and should be incorporated into the plan as well even if its cost-benefit may be hard to quantify. We all know that there are a lot of "unknown unknowns" here as well as simply a lot of "hard-to-quantify known unknowns." My goal is merely to propose a comparison "yardstick" by which I think any alternative hardening approach would need to measure up before we can say it should be used (alone) as a *replacement* for software implementation diversity. If we can't find solid analytical or empirical evidence that some hardening process P reduces one component's failure probability at least exponentially in added investment, then process P shouldn't *replace* software diversity measures. Perhaps we should still implement hardening measure P, *in addition* to diversity measures - it just shouldn't replace diversity.

Edited by Bryan Ford 7 months ago

Emilia Nunes @Emilia.Nunes · 7 months ago Developer

@Bryan.Ford thank you for your detailed answer. I would like to ask you a few additional questions:

- What are the downsides for implementing a software for a least one of the control-components?
- What are the requirements for building such software?
- What kind of know-how is needed in order to build such software?

Bryan Ford @Bryan.Ford · 7 months ago Developer

Hi Emilia, I'm not sure if I understand your first question about the "downsides" - can you clarify? Are you asking about the potential downsides (risks? costs? other?) to a potential third-party developer of an independent implementation of the control component software? (Which would presumably need to be different from the e-voting system provider, who must produce a control component implementation in any case?)

The main "bottom-line" downside, as I see it, is the cost, which I tried to capture above even if in simplistic fashion. Developing n fully independent implementations of the same specification may cost up to n x what just a single one would cost.

If we would like to analyze the other business, legal, or logistical cost-benefit factors to a potential developer of an independent implementation of the control component software, then that's probably a bit complicated and not exactly my area of expertise but I could hazard some guesses. :) A third-party developer of an independent control component implementation might be reluctant to accept such a role because they don't get to design the overall system and hence don't have a "lead role" in the architecture or implementation, which might imply lower (perceived) opportunities to capture market share and make future profits off an investment. So a third-party control component implementation provider might be less willing to "subsidize" their effort in anticipation of future profit potential, and rather demand the full development and validation cost more-or-less directly.

A third-party control component implementor might also perceive legal or repetitional risk factors: e.g., being exposed to embarrassment or worse if critical bugs are found in their implementation - even if a diversity of control components successfully prevents that critical bug from rendering the system as a whole vulnerable. So a third-party control component implementor might need or want to "price in" this perceived risk as well in some fashion. (But I don't have any particular reason to believe such risks would seem worse for a third-party implementor than for the main e-voting system provider.).

Another issue that has been brought up a few times is the downside in "complexity" of having multiple control components. That is

another area where downsides could lie. Perhaps other experts could elaborate on the complexity factors they're worried about?

The most obvious forms of additional "complexity" I think are already captured in the above (albeit simplistic) cost-benefit model, in that producing 1 specifications and 4 independent implications is probably going to involve writing, debugging, and validating a bit less than 4x the total amount of code that would be required to produce 1 specification and 1 implementation. The same goes for operational deployment factors: planning out four different secure locations for four different sets of equipment, based on four different sets of OS and hardware combinations, purchased through four separate vendors or supply channels, and set up and staffed by four independent pairs of administrators, is again going to require dealing with somewhere about 4x the total planning complexity, deployment costs, and operational costs as just running one non-redundant and non-diverse control component. (But maybe I'm diverging into other discussion blocks here, sorry.)

But of course "complexity" may also refer to design and implementation complexity factors related to having support for trust-splitting across multiple control components in the first place, as opposed to simply having one fully-trusted control component. Let's take a quick look at some of these factors (I'm sure I'm forgetting plenty that others might want to add.)

There can be cryptographic design costs, in principle: e.g., the need to use cryptographic techniques that enable private keys and sensitive operations like shuffling to be split across multiple control components. But practically all of the state-of-the-art discrete-log-based cryptosystems that e-voting systems tend to rely on have this capability more-or-less built-in "anyway", whether or not a particular system design chooses to use it. If end-to-end verifiability is a goal, then you'll need a zero-knowledge proof of a shuffle, regardless of whether you just have one control component producing one shuffle proof or you have four control components producing four shuffle proofs. The need to have that shuffle proof in the design at all, and implement it correctly in the system, is where the main complexity is - and that doesn't seem easy to avoid without giving up entirely on verifiability. So frankly I don't see much design complexity cost to supporting multiple control components in the form of the amount or complexity of crypto tools needed in an e-voting system that's supposed to be verifiable.

There can be protocol-level complexity costs, as well, in that there may potentially need to be (for example) protocol machinery enabling the control components to coordinate with each other to satisfy client requests and such, whereas a single fully-trusted control component needs no protocol machinery to "coordinate with itself". However, in my experience, this complexity factor is heavily dependent on the design and can easily be kept rather small if the overall system architecture is well-designed. Consider for example the nice "Individual Verifiability" summary diagram. In this architecture, the control components mainly interact - in parallel - directly with clients, not with each other. There is probably management software, etc., that also needs to interact with the control components, but that software - and the protocols it needs to interact with the control component(s), whether one or several, might be pretty similar either way. So it might be fairly easy to avoid the control components ever needing to have to communicate with each other directly, and thus to avoid incurring the protocol complexity cost of supporting that interaction. Much of the "trust-splitting" cryptography that e-voting systems tend to use works fairly naturally in this parallel fashion: e.g., where the client is responsible for obtaining shares of something from each control component and combining them into the joint cryptographic whatever.

A possible exception to this rule - and here I'm diverging into the topic of Discussion D, sorry - is the mixing/shuffling part. This is because shuffling in classic "cascade mix" designs as e-voting systems use fundamentally needs to be serialized: one mixer first shuffles and re-encrypts the whole batch of votes and produces a proof, then the next mixer takes the first mixer's output and does the same, etc., one after the other. There's no way around this other than using a completely different approach to anonymization of votes (e.g., DC-nets, which I've done a lot of work on, but which I definitely do not propose is the thing needed for e-voting. :)). So for mixing it seems more likely that supporting multiple control components (as opposed to just one) may add more protocol design complexity cost, to support this forwarding of cipher text batches among control components. But even this design complexity cost can be mitigated with careful design, which in my experience is not hard to achieve. For example, a common design pattern is to keep the mixing servers technically purely "passive", just taking orders from an untrusted front-end server of some kind, which directs the vote batches through all the mix servers. In this design, the mix servers only need to talk a single simple protocol - namely the one with the front-end - and still don't need to talk directly with each other or even know directly of each other's existence. This is just good system design practice.

I'm sure there are plenty of other areas where supporting multiple control components, as opposed to just one, might arguably add complexity to the system design or implementation. But most of the obvious ones, when I've looked at them more closely, seem not too serious or to be subsumed in the simple cost/benefit analysis above. But maybe other experts would like to bring up other areas of complexity costs that we should discuss.

Bryan Ford @Bryan.Ford · 7 months ago

Developer

And sorry Emilia, forgot to answer your other two questions, about the rights and know-how for building such software... I think that every team producing an implementation of a control component definitely needs to have at least one experienced cryptographer participating directly or, at minimum, reviewing all the code as it is written. And the whole team should preferably be experienced in systems and software engineering with strong code-review and "implementation-for-verifiability" disciplines.

But a lot also depends on the specification they're writing the implementation to. One of the advantages of having (requiring) multiple independent control component implementations, as others have already pointed out, is that it forces multiple teams to analyze the specification closely and deliberate on any perceived issues or lack of clarity. Not only having that separate specification in the first place, but also having it extensively scrutinized by multiple independent teams trying to understand and implement it, may be seen as a "complexity cost" in that it takes lots of discussions and deliberations between the teams to identify and fix problems, rewrite everyone's code as they are fixed, etc - but from another perspective that is not just a cost but an extremely valuable benefit, in that the result will likely be far more clearly-specified and well-understood by all the implementors, hopefully the actual code will be able to follow and correspond to the specification pseudocode more directly and clearly, etc. So the "complexity cost" of multiple control components in terms of deliberating over specification issues and fixes throughout the development process, I think, is really just a "cost of quality" that should and needs to be paid regardless, but it's just easier to "cut corners" and

avoid paying some of that cost (and correspondingly compromise quality) if you're one provider producing a specification and one-and-only implementation together.

Christian Folini @christian.folini · 7 months ago Maintainer

Wow. That's an exhaustive answer to this question and the followup one by Emilia. Thank you Bryan.

Bryan's statement is very clear that investing the money into additional software brings the best return on investment and he sees no hardening technique that could bring similar (exponential!) benefits.

I take it he would use the same arguments when answering the questions B, C and D (which is fair I think).

For A, Bryan does not merely see added complexity when implementing additional software, but a better specification emerging during the implementation process. So it's more a "cost of quality" that should be paid.

I think it is not possible to address question B (-> independent operating systems) in the same way as the know-how, infrastructure and processes to maintain different operating system really brings substantially higher costs (and a substantial availability problem when you - e.g. - rely on a single OpenBSD expert).

Therefore we can not simply extrapolate this answer A to B, C and D and we definitely need separate answers from other experts for the other questions.

We saw a qualified majority for Bryan's position in the questionnaire already (see summary of 2.1.4 for example). However, there were differing opinions in the questionnaire and I'd like to know if Bryan's extensive explanation convinced the opposing camp that would rather prioritize one piece of software and harden it extensively. I count @Stephane.Adamiste, @Sergio.Alves.Domingues, @Oscar.Nierstrasz and @Uwe.Serdult among this group. It is likely that BFH (-> @Eric.Dubuis, @Rolf.Haenni, @Reto.Koenig and @Philipp.Locher) as well as @Ulrich.Ultes-Nitsche disagree with Bryan's position too based on the questionnaire. At least for A.

In other words: Is this really all there is to say here or are there arguments in favor of the "single hardened software" approach - for this question A as well as the other questions in this discussion block.

Or maybe everybody is just blown away by the vibrant energy of Bryan's response.

David Basin @David.Basin · 7 months ago Developer

I support Bryan's response.

A useful read here is Ken Thompson's Turing Award lecture "Reflections on Trusting Trust" (https://dl.acm.org/doi/pdf/10.1145/358198.358210). He shows how compilers can easily build in backdoors into systems and even into the compilers themselves. His conclusion is "The moral is obvious. You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code." So diversity has the advantage that you are more likely to be using non-malicious APIs/Build-Environments/hardware/... for at least one of your (sub)systems. This is an additional point beyond resilience to bugs leading to zero-day exploits.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

I also support Brian's response.

However, there is something about the use of multiple control components for establishing individual verifiability I don't get yet.

Why they are necessary for mixing is clear: We just don't have a better way to guarantee reasonable levels of vote privacy: Current system designs always require multiple mixing nodes because 1) one can argue that steps have been taken to minimize the probability of the EMB breaking vote privacy (as long as one mixing node is honest and keeps its permutation a secret), and 2) rhetorically one can use this argument to present a compelling argument to gain the public's trust (the vote privacy is not at the same level compared in polling place voting, but more in line with voting by mail). I'd like to note that we actually do not know very much about the probability of a mixing node revealing or not revealing a permutation.

But is it really true that we don't have anything better than control components to ensure a trustworthy return code mechanism? What puzzles me is that to achieve individual verifiability, the voting protocol itself is not required to be verifiable, but the control components are required to be trusted? Related, a broken control component seem to be able to take the entire system down: A faulty return code share would (with non-negligible probability) make a voter (perhaps all voters) receive the wrong return code? Doesn't that mean that you want to have as few control components as possible? Note that this is different in mixing. If one mixing node doesn't work, or the zkps don't check out, one could skip this particularly mixer and move on to the next.

Christian Folini @christian.folini mentioned in issue #14 (closed) 6 months ago

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

@Carsten.Schuermann you asked lots of questions in one paragraph!

> But is it really true that we don't have anything better than control components to ensure a trustworthy return code mechanism? Are you asking whether it is possible to generate return codes in a verifiable way without any trust assumptions? My guess is that the answer is probably no, because it boils down to decryption. The question of whether it is possible to do individual and universal verifiability in a better way, without trust assumptions, e.g. using a public bulletin board, is a separate question (to which I think the answer is probably yes), which I think we discuss later.

> What puzzles me is that to achieve individual verifiability, the voting protocol itself is not required to be verifiable, but the control components are required to be trusted? Related, a broken control component seem to be able to take the entire system down: A faulty return code share would (with non-negligible probability) make a voter (perhaps all voters) receive the wrong return code? Doesn't that mean that you want to have as few control components as possible?

Let's separate out two different attacks:

1. Individual Verifiability Failure. The voter receives the correct return code though their vote has not been properly recorded.
2. FUD Failure. The voter receives a faulty return code though their vote has been properly recorded (or at least, their client sent the right vote).

I spent a fair bit of time reading through Scytl's protocol for distributed vote code generation, and I believe it attempts to defend against both (though I'm not convinced it does so successfully).

Attack 1 is the main motivation for requiring multiple authorities, of which at least one must be honest, hence the wish for multiple control components. Assuming that return-code generation boils down to decryption, we want the decryption key to be shared among multiple components so that they can't all pool their keys, decrypt every possible return code, and decide to send a deceitful one back to the voter. The Scytl/SwissPost protocol went to considerable lengths to attempt to guarantee that only one code could be decrypted, and only with the participation of all four control components.

You're probably right that this increases the likelihood of at least one being bad, and hence perpetrating attack 2, but that seems less bad than attack 1.

So *if* this is the design style we choose, then we want to have lots of independent control components. Though I agree it's not clear that this is the only, or the best, style of protocol.

[EDIT: folini: formatting: "Let's separate ..." appeared as quoted.]

Edited by Christian Folini 6 months ago

Bryan Ford @Bryan.Ford · 6 months ago Developer

Good points. What @Vanessa.Teague calls a FUD Failure I would classify as a specific type of denial-of-service (DoS) attack. Which is certainly an important class of attack that we need to consider and worry about. We definitely have evidence of nation-state adversaries basically trying to sow chaos and FUD, e.g., just trying to give democracy a bad reputation and make people think it doesn't work at all, as opposed to necessarily trying to manipulate votes or election outcomes numerically.

But in an "anytrust" system with an n-of-n threshold - where you have to trust only that "any one" control component is uncompromised - this fundamentally means that you must trust *all* n components for liveness, which yields many potential DoS attack vulnerabilities. For example, if an adversary can just kill the power to one control component for an extended period, or can cause both of the only two sysadmins with access to that control component to go missing (even just temporarily), or can use a BGP routing blackhole or similar network-level attack to cause many/most clients to be unable to communicate with one of the control components, or ..., then you haven't compromised election integrity but you have compromised liveness/availability. And this alone might well satisfy the "sowing chaos in elections" aims of some realistic nation-state attackers.

As usual, there are various ways to mitigate these types of DoS risks. I'm pretty confident that the Swiss voting system design(s) could be modified to support t-of-n threshold operation with t < n (e.g., 3-of-4), but that would definitely add significant design complexity that would have to be weighed and justified carefully against its costs. For example, the control components could use Shamir secret sharing appropriately for cooperative management and decryption of verification codes for example: in principle this shouldn't be particularly difficult, but doing anything (correctly) with Shamir secret sharing requires more complex setup protocols and more complex calculations many other places, rather than just adding/multiplying shares together directly. t-of-n threshold operation where t < n also would likely increase complexity significantly in the mixing phase. Even though @Carsten.Schuermann is right that this would in a sense just require skipping any (one) failed mixing node, this presents us with major secondary challenges like making sure all nodes are agreed about which mixing node(s) have failed and what the "definitive" mixing order turns out to be (even after the fact), and ensuring an adversary can't successively produce two different shuffles that both appear legit but are different and might cause confusion downstream in counting... Basically, doing t-of-n mixing in a truly robust fashion seems to require some form of Byzantine consensus to agree robustly on which specific t mixers are or have participated and making sure that one and only one such mixing sequences is agreed by everyone. And as we know, Byzantine consensus is complex and generally a real pain wherever it comes up.

As a result, I personally think that in this respect the way the Swiss voting systems already handle availability and DoS risks is probably best, at least for now: namely just make sure all four control components (and all the mixers etc) are each individually highly-available (redundant server hardware, redundant power supplies, redundant network links, redundant admins, all that), using the standard "off-the-shelf" mechanisms and practices to achieve high availability. Thus, an "anytrust" n-of-n design effectively

protects integrity and privacy at a higher level while protecting availability at lower level (in the hardware/software/processes for managing each node), which I think is a pretty good separation. So given the complexity tradeoffs with n-of-n versus t-of-n designs I think the Swiss system is already pretty much at the state-of-the-art "sweet spot". But reasonable people might differ on that.

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

I must admit that in my first reading of the various documents, the architecture of the control components was not clear to me, hence my skepticism about the benefits of multiple implementations. If indeed it is enough for one of the four systems to succeed, then the benefits are enormous, as Bryan Ford has clearly expressed. Redundant systems are a standard way to achieve high reliability, but only where all systems are truly independent. In case of disagreement, "voting" is often needed to resolve differences between disagreeing systems, but that appears not to be an issue here, since only one system needs to succeed. I withdraw my earlier comments about the increased attack surface as they clearly do not apply here.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Oscar. I was really hoping you would chime in and put your statement from the questionnaire in perspective after reading the other responses. (Not because I hoped you would revoke it, but because your opinion really matters here.)

Edited by Christian Folini 6 months ago

Rolf Haenni @Rolf.Haenni · 6 months ago Developer

@Carsten.Schuermann

> But is it really true that we don't have anything better than control components to ensure a trustworthy return code mechanism? What puzzles me is that to achieve individual verifiability, the voting protocol itself is not required to be verifiable, but the control components are required to be trusted? Related, a broken control component seem to be able to take the entire system down.

In CHVote, we realise a trustworthy return code mechanism using an oblivious transfer (OT) protocol, see Sections 5.3 and 6.4.3. of https://eprint.iacr.org/2017/325.pdf

This works with any number n of servers, i.e. privacy is also guaranteed with a single server. In the multi-server case, the technique could easily be generalized to obtain the return codes in a threshold manner, but this is not (yet) implemented in the CHVote protocol.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Rolf. Good to have BFH in this thread as well.

Do you also have an opinion on Bob's question above? Should he buy a new software for the control components?

Or do you mean to say, that with your alternative approach, there would not be a need for multiple control components.

Florian Egloff @Florian.Egloff · 6 months ago Developer

Again, slightly orthogonal to the "technical" questions, which others have expanded on in detail.

@Bryan.Ford said:

> Good points. What @Vanessa.Teague calls a FUD Failure I would classify as a specific type of denial-of-service (DoS) attack. Which is certainly an important class of attack that we need to consider and worry about. We definitely have evidence of nation-state adversaries basically trying to sow chaos and FUD, e.g., just trying to give democracy a bad reputation and make people think it doesn't work at all, as opposed to necessarily trying to manipulate votes or election outcomes numerically.

1. I would ask: do we discuss these aspects in a different block separately or here? If here, I would frame the "FUD problem test" as: Given an allegation of cheating/misconduct, is the remediation trusted and trustworthy by the population (voters, non-

voters, non-eligible residents)? Given multiple threat actors deliberately injecting doubt&mistrust into and around the voting process, is the outcome still trusted and trustworthy by the population? I hope we will discuss the trust and trustworthiness issues in-depth, here, or elsewhere.

2. Regarding independence: we should probably also discuss, in how far the separate voting channels are independent of one another. Is the distributor of physical mail (in CH the SwissPost) the right partner to manage the online systems, given the significant trust placed into the distributor in the physical remote voting channel? When thinking about single points of failures and cross-over trust problems between channels, this came to mind.

3. Along the same lines: is it a problem or strength that the operator is at the same time the distributor of the authentication and return codes? I.e. do we have to "trust" the operator anyway, and hence, we mitigate an additional point of failure when they are unified, or is there an advantage in keeping them independent of one another? My intuition is that for operating the system, the operator does not necessarily need to know who is assigned which codes. This is something that the distributor could potentially learn (by opening the envelopes). What do people in the group think?

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for bringing this up, @Florian.Egloff .

(1) This is a good reminder to include this question in the discussion we planned for question 2.3.1 of the questionnaire as well as 2.3.2 and 2.3.3.

The ideal procedure a voter should be taking is: Inform the authorities the return code was wrong and then use one of the physical channels to submit his / her vote.

(2) and (3) these weaknesses of the process are existing today with the physical voting channels already. Introducing a third channel is not aggravating the situation, I think. It might even bring a minimal mitigation.

It is probably worth to look into Killer 2019 where this is discussed at some depth.

So far, there is no plan to discuss this in the dialogue. However, if there is a substantial call among the experts to look beyond online voting, namely with the trust assumptions for the physical transport, then FedCh is ready to put it on the table (after consulting with the cantons, obviously).

So if you think this should be discussed, then please make yourself heard.

[EDIT: Formatting]

Edited by Christian Folini 6 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

On (1) I look forward to the discussion.

@christian.folini ;

> The ideal procedure a voter should be taking is: Inform the authorities the return code was wrong and then use one of the physical channels to submit his / her vote. (2) and (3) these weaknesses of the process are existing today with the physical voting channels already. Introducing a third channel is not aggravating the situation, I think. It might even bring a minimal mitigation. It is probably worth to look into Killer 2019 where this is discussed at some depth.

Killer & Stiller 2019 do raise interesting points. Two thoughts come to mind:

1. Killer & Stiller 2019 point out the existing digital risks in the current physical voting process. These are problematic for two reasons: 1. if the physical voting process is supposed to serve as a "backup strategy" for the case of a "failed" internet vote, the independence of the voting channels would be important. 2. If the voting channels are exposed to the same threat actors, which in a digitalized physical voting/tallying/transmission of results procedure is the case, one would have to consider the risk profile not only to the internet voting system but also the physical system.

2. Actually, Killer & Stiller 2019 do not analyse how the risks would change if the operator of the remote internet vote is the same as the distributor of the remote postal vote. Internet voting is supposed to reassure voters that their vote was cast-as-intended, recorded-as-cast, and counted-as-recorded. If the distributor is controlling both the system and the delivery, then it has an increased potential to undermine the vote.

Christian Folini @christian.folini · 6 months ago Maintainer

Ah, sorry. I did not mean to express that Killer/Stiller talk of a minimal mitigation. I should have started a new paragraph between the two sentences. I meant to say that this article is one of very few that discusses voting by mail from a threat modeling perspective

and that looks at digital systems in the physical voting process.

However, this topic is not the focus of our talk here. If you want it discussed nevertheless, we need more experts saying so.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I would like to second a call to include physical transport assumptions, e.g., the assumption that the postal service is a perfectly reliable and trustworthy black box, somewhere on the discussion agenda.

While I won't pretend there's a reasonable escape from that assumption in the current-generation e-voting design, to me it's a huge lingering problem and risk area in all sorts of ways and fixing it at least needs to be discussed and hopefully put on the longer-term roadmap.

I think these issues become doubly important if a significant part of the Swiss e-voting system's user base and interest/support base is from Swiss expats living in other countries around the world, who need to cut the postal round-trip latency in order to vote reliably at all. In the case of these voters, we're not talking (only) about how much you want to trust the Swiss postal service; we're also talking about how much you want to trust the postal services of every other country in the world where Swiss voters live.

Florian Egloff @Florian.Egloff · 6 months ago Developer

So @Bryan.Ford, @Tobias.Ellenberger, and @Adrian.Perrig seem to agree that the physical aspects should be discussed somewhere.

@christian.folini 's answers re DoS mitigation also point to the important nature the physical voting process seems to take, namely, as a risk mitigation for the internet channel. If this is the case, in my opinion, you cannot disregard the security issues with the physical process, as if you want to use it as mitigation, the strength of that mitigation can only be assessed with an understanding of the physical voting process. Please clarify whether this is in or out of scope.

There is one important point I would like the FedCh to take on board from this: Capable threat actors that could act against the internet voting process, could also act against the digital elements in the physical voting process. This means that treating those channels independently fails at addressing the threat wholistically. Consequently, risk mitigations should be taken in both, if you want to address the threat to election/referendum security.

Bryan Ford @Bryan.Ford · 6 months ago Developer

Catching up after a while, I just wanted to highlight this phrase from @Florian.Egloff:

> Capable threat actors that could act against the internet voting process, could also act against the digital elements in the physical voting process.

Extremely well-stated and I couldn't agree more. A lot of people, including many experts I think, vastly under-appreciate how many potentially-vulnerable digital elements there are in ostensibly paper-based voting processes.

Florian Egloff @Florian.Egloff mentioned in issue #11 (closed) 6 months ago

Reto Koenig @Reto.Koenig · 6 months ago Developer

(We had quite an intensive discussion about the discrepancy of our response compared to the response given by @Bryan.Ford for this question. And we must admit that there is a non negligible probability that we are wearing somewhat rose-colored glasses, regarding this aspect.)

The answer we have provided in the questionnaire was based on the following assumptions to be true:

- There is a detailed specification of the cryptographic voting protocol which has been proven correct and verifiable, and which can be implemented with no additional cryptographic knowledge.
- There is an implementation that is directly derived from this specification, and all cryptographically relevant operations can be mapped back to the pseudo code of the specification.

Then, we believe that a single open (white-box) implementation that has been audited and scrutinized by many experts might be less distractive than if there were multiple dissimilar implementations of the same specification available. (Whereas distractive is meant in a way that there might exist too many implementations and too few experts)

However, we are well aware of the practice of using dissimilar components in soft- and hardware for mission critical components (Airplanes, nuclear plants,...). The target there is set on robustness on every layer even against unintended software bugs on implementation level (usually black-boxed). The system must remain in a stable state even if a minority of components is flawed. But that is not the case in our e-voting system, where we are focussed on the correctness of the result by providing a verifiable

implementation based on a verifiable specification using independent implementations of universal verifiers based on the proven verifier specification (White-Boxed).

But still, we agree with Bryans statement to provide separate implementations completely independent from each other, only based on the given specification. Be it white-box or even black-box due to the following reasons:

- Dissimilar implementations can be used to check if they provide the same result when run in a deterministic way. We see that as a very strong tool to audit the correctness of the implemented protocol steps.
- Running dissimilar implementations during election period certainly increases the costs for a successful attack on compiler/library level.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you ever so much for this careful and balanced response. It's easier to understand your answer(s) in the questionnaire now. I admit I was getting a bit uneasy in absence of BFH in this thread.

Christian Folini @christian.folini · 6 months ago Maintainer

Reading this a 2nd time, I am no longer entirely sure about the answer you want to give to Bob.

There is a way of reading your statement that goes as this: We have written a set of specifications that is of a very high quality and it avoids all wiggle room for the developers. Therefore you do not need multiple implementations: Bob, use our protocol and you do not need independent control components.

Or the alternative reading: We believe the key is to have very high quality specifications avoiding all wiggle room for the developers. Yet, we are aware that online voting is a complex piece of software, humans make errors and even if you get vast public scrutiny on your implementation(s), there is a substantial advantage in developing and maintaining alternative implementations even for our protocol: Bob, you should invest your resources into independent control components.

So which one is it?

Reto Koenig @Reto.Koenig · 6 months ago Developer

OK: Let us nail it. Imagine Bob is an employee of a cantonal administration. It is his job to convince himself and others that internet voting is sufficiently secure....

Now, let us further consider Bob to have only limited capabilities in analyzing cryptographic protocols and implementations.

If there is a dispute about the quality of the specification and its given implementation between system vendor and scientific experts, then we highly advise Bob to request for another independent implementation of the control components relying exclusively on the given e-voting specification. This way, Bob is not required to judge about the dispute which would be very difficult for him with his forementioned limited capabilities.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for following up, Reto. Much clearer now.

Christian Folini @christian.folini changed title from **Discussion A - Independent Software for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 1A - Independent Software for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Christian Folini @christian.folini changed title from **Discussion 1A - Independent Software for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 2A - Independent Software for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Tobias Ellenberger @Tobias.Ellenberger · 6 months ago Developer

good answers and overwhelmed by the energy of bryan ;)

from a technical point of view, agreed with the general direction regarding the recommendation to Bob to invest the money in

additional software if possible. Provided - besides the assumptions already described - that the additional solution to be used meets the same security standard as the solution already in use.

From a business perspective, the comment that a new software can be expected to bring additional effort and complexity, even if the additional software is managed by an external entity. However, the additional complexity is worth it in terms of security gains. Depending on how many "additional funds" are available, the advice may be to use the money to mitigate the risks associated with the existing software. In my opinion, this is conceivable for smaller cantons, where the funds may not be sufficient despite additional resources.

> However, this topic is not the focus of our talk here. If you want it discussed nevertheless, we need more experts saying so.

think this can be relevant and interesting

Edited by Tobias Ellenberger 6 months ago

Adrian Perrig @Adrian.Perrig · 6 months ago Developer

Adding to what @Bryan.Ford pointed out about network DoS attacks:

> "or can use a BGP routing blackhole or similar network-level attack to cause many/most clients to be unable to communicate with one of the control components".

This is a point I wholeheartedly agree with: we need to address network-level attacks. Fortunately, methods exist to achieve this even in a multi-provider network, for instance preventing network kill switches by external entities -- this is to ensure network connectivity with minimal assumptions. Note that this is challenging to achieve, as an adversary could for instance attack the hardware switches of network service providers.

Nevertheless, such risks can and should be addressed, in particular to avoid problems with backlash against electronic voting in case part of the nation could not submit their vote.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Adrian.

Before we dive deep down into DoS (talking DoS is one of my favorite discussion topics btw), there is an important procedural mitigation technique in place that guarantees that most if not all voters can submit their vote even when a DoS is making the online voting service unavailable. I think it is important to keep this in mind:

The electronic urn is closing on Saturday. This means that if a DoS hits, you can still submit your vote physically on Sunday. In the case of foreign voters, it's of course too late, but they are instructed to vote immediately after receiving the printed material.

This does not rule out a DoS of course (so many interesting options...), but it makes it rather unattractive, which is often enough to make sure a DoS is not happening.

Of course, this does not protect the control components at all. If you are able to DoS them one way or the other on Sunday afternoon, then the tallying process will be affected.

Adrian Perrig @Adrian.Perrig · 6 months ago Developer

Perhaps it's better to discuss this in the segment of printing offices, however, it may be good to bring this up here as there was mention of physical transport assumptions.

As for the physical transport assumptions, and also to mitigate trust requirements into the printing office, would it be possible to send multiple (I guess two is probably the max number of items that can reasonably be sent) items to the voter? Then the voter combines information on all the items received. In that way, the effect of compromise of a single printing office would be mitigated. (An interesting approach may be to send one item via digital means, e.g., SMS or email, although this may open up a Pandora's box of new issues.)

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

This is an interesting idea. It is not totally new, but so far it was always ruled out due to usability concerns. We plan to get back on the usability discussion or you could bring in this idea into the print office discussion (even if the questions there do not leave much room for additional ideas).

Bryan Ford @Bryan.Ford · 6 months ago Developer

I agree that "multiple physical items/tokens" approaches are potentially worth exploring, but I'm skeptical about about them as well not just for usability reasons but also diversity concerns. By the usual independence/diversity principles, sending multiple physical tokens of some kind is a security benefit only to the extent that those tokens are produced independently, and sent to the voter via substantially independent channels, so that there's a reasonable chance that a compromise in one path won't immediately imply a compromise in the other. This seems to mean that the multiple tokens or items would have to be produced by independent printing authorities (maybe not so hard), but maybe more difficult, also sent through different postal or delivery services. Most countries have only one "postal service" per se. I guess one delivery channel could be SwissPost, the other DHL or other express delivery companies - but the expense and logistical complexity seems likely to explode.

Sergio Alves Domingues @Sergio.Alves.Domingues · 6 months ago Developer

Our answer (favoring the use of a single properly coded and thoroughly reviewed implementation over a variety of different ones) was mostly based on the idea that – if both could not be simultaneously achieved – proper implementation and thorough review could provide more trust than mere diversification. As mentioned, we transposed the idea that, most of the time, relying on a well-known and properly reviewed crypto library is better than trying to create a new one (or by extrapolation use two that are poorly reviewed and rely on diversity).

We do not see diversification as something bad, however in the context of tasks like the proper implementation of complex specifications (e.g. crypto) we had the feeling that focus should first be directed to proper review and that a single good implementation seems better than a few “average” ones combined.

That being said, this was just based on a simple principle and not on a formal model or even rough estimation of cost/effort benefit. As @Bryan.Ford demonstrated, this assumption seems to be wrong (i.e. we may have a bigger benefit for the same cost by investing in diversification) at least against the scenario of a malicious actor whose task would be to compromise all control components.

Considering attacks requiring the compromise of a single control component (e.g. attacks falling into the “FUD Failure” category expressed by @Vanessa.Teague) diverse components appear to imply a broader attack surface. Both impacts may however not be comparable.

Stephane Adamiste @Stephane.Adamiste · 6 months ago Developer

Although I am pretty convinced by the arguments of people advocating for multiple diverse components, I would like to contribute to the debate by questionning the need for improved security in that area in particular. It seems to me that implementing several independant/different components for verifiability raises the exact same issue as the one those components are meant to fix, due to the complexity of implementing them: How to prove that those components do their job properly? Aren't we creating an infinite loop here? The security assurance effort to dedicate to those additional components is far from negligible, and will not satisfy skeptical people anyway. The more different components, the bigger the effort.

@Bryan.Ford provided rationales that seem to show that the investment/benefit ratio for multiple components is interesting from a security perspective, but another interesting aspect to consider is whether the corresponding risks (e.g. the ones mentioned by @Vanessa.Teague) justify such advanced means to reduce them compared to other risks faced by the e-voting. In my view, this parameter should be included in the discussion (not sure whether this thread is the right place though...).

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

@Stephane.Adamiste: "How to prove that those components do their job properly?"

This is the role of the verifier. The control components have to prove to the verifier, that they are doing their job correctly and the integrity of the vote is guaranteed as long as at least one of the CCs is not compromised.

Christian Folini @christian.folini · 6 months ago Maintainer

Several experts in this thread stated they are uneasy with the trust assumption that the physical transport by Swiss Post is a perfectly secure transport channel. Therefore, they would welcome a discussion about this topic.

While we appreciate the interested, we are also reluctant to shift our focus away from electronic voting to problems of the voting process in general; thus problems that also affect the physical voting channels, which is a very wide area.

But there is a middle ground between a fully blown discussion and ignoring the problem. See Discussion 99A - Voting Security Outside of Online Voting.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you all for contributing to this very interesting thread.

We have read many, many interesting arguments now. Namely here in 2A, but also in 2B, 2C and 2D, although to a lesser extent.

Now let me see if we are able to sum it up.

Given that 2B-2D are very close to 2A and 2A got most of the comments anyway, I will try to do a single summary for all four questions.

Since not all questions have been covered in the same detail (hat tip to @Bryan.Ford), I will need to extrapolate a bit. If I am stretching this too far, then I welcome your feedback and alternative wordings.

Software diversity is something that is very obvious to IT people, but we are going to have a hard time explaining it to the non-techies. I will thus try an ecosystem metaphor that is easier to explain, but still carries the same arguments and conclusions. I'm not sure this will work, but a draft summary is meant as a work in progress, so this need not be perfect on the first attempt.

Please sit down, this is going to be a long read:

Diversity in IT Security

It is a common understanding in the IT industry, that diversity leads to systems that are more resilient to attacks. The effect is very similar to a biological ecosystem, where monoculture is cheaper to maintain. But successful attacks in the form of fungus or any other form of pest is going to have a more devastating effect in a monoculture. It is true that investing more resources into the protection of the monoculture will have a positive effect. However, using the same amount of money for diversification, has a substantially better cost-benefit ratio when looking at the probability of a successful attack wiping out the whole ecosystem. And given the design of the internet voting system, an attacker has to wipe out all four control components and the verifier to manipulate a vote without being noticed. Therefore, the system is designed to get the biggest benefit possible out of diversity.

There has been a lot of research into this question for IT systems. The conclusion has been that diversification brings exponential cost-benefits while more hardening and more scrutiny will bring less and less security gain as investment increases. This follows a pattern of diminishing returns.

In 1984, Unix legend Ken Thompson received the Turing Award and held an acceptance speech that has proven to be a timeless classic. Published under the title of "Reflections on Trusting Trust", Thompson came to the conclusion, that "No amount of source-level verification or scrutiny will protect you from using untrusted code." Diversity is a relief for this problem, namely for Swiss internet voting. This is because it forces an attacker to exploit all four control components in parallel. This is (a) very expensive for the attacker and (b) could lead to the attack being detected, which is very risky for an attacker.

Hardening and Scrutiny Still Important

This does not mean, that hardening and scrutiny are futile. There still needs to be a high level of hardening and scrutiny, not the least because public trust will depend on these audits. Also formal verification methods bring a high level or reassurance to those parts of the system where they can be applied. This also contributes to the public trust. However, there comes a level of security, where diversification brings better benefits, namely in the long run, for unexpected attacks and for those parts of the system where formal verification can not be applied.

Downsides of Diversity

Admittedly, diversity also comes with downsides. The costs of the independent implementation is the most obvious one. This includes the coding, the documentation, the integration, etc. At first sight it also includes a better specification, since independent developers will need to be able to write their code based on the specification alone. A single implementation can be developed based on a specification that has some ambiguity. But with a second, independent implementation, the specification is being validated by 3rd party developers, the ambiguities will surface and will have to be cleared out. This in turn leads to a better specification. So the additional costs that has to be put into the improvement of the specification (if any) is in fact an investment into the quality and the security of the system.

It would be useful to enforce conformance testing on the independent implementations. This would help in situations where shortcomings of the specifications were not noticed during the development.

Operation is also more expensive with multiple independent systems; namely when you also look at potential different behavior and how they ought to be reconciled as part of the operation. The latter however is but a small part of the operation, so we can still assume that the costs are more or less linear, while - as pointed out above - the benefits are exponential. So it is a worthwhile investment also in this regard.

There is also a bigger attack surface with multiple systems. But the more independent the control components are, the less can an

attacker profit from one exploit when attacking the next one. The weaknesses are thus limited to individual control components and they no longer harm the security of the overall system. Of course, operation has to guarantee proper separation of the control components, so that one system can not be used as a bridgehead to attack the next one.

Diversity as Core Requirement

All combined, the benefits of the diversity outweigh the costs by far.

So diversity is a core requirement and ideally every control component would be completely independent in every possible way. Several experts made it clear that they require a high degree of diversity to start to consider the system as trustworthy. Unfortunately, implementing four entirely separate control components is not feasible in real world. But there have to be rules to establish a minimum of diversity as a hard requirement.

Priorities

Even highly skilled attackers will not attack all levels of the system with equal ease. It seems likely that the application software of control components would be attacked first, since it is a custom implementation of software with limited public scrutiny. Afterwards, more widely used components, 3rd party components, would be attacked. The experts expect a powerful attacker to possess the means to exploit widely used software components (libraries, the Java virtual machine, operation systems and firmware, but also compiler etc.) and possibly also hardware components (CPU, mainboard, controllers etc.) However, exploits for widely used software and hardware components are very expensive to obtain and using them in an attack where they could be discovered makes it unattractive to use them. So, as a potential consequence, the OS and other 3rd party components would only be attacked if it brings an advantage over attacking all four control components together.

As a consequence, diversity for the control component application software must have the highest priority. This is also where the costs are the highest since it demands a separate implementation.

With the libraries, OS, firm- and hardware components, the situation is different, since these components do not have to be developed. Here the costs boil down to operation to a very wide extent. Therefore, the costs of the diversity are expected to be much lower in this area, especially for a system provider that has operation procedures in place for more than one operation system and for more than one type of server hardware already. Diversity is thus a cost-effective means of defense beyond the control component application software.

Components that Deserve Special Treatment

Two or three items are special and they deserve separate coverage: It should be required to use a different programming language and a different compiler for different control component implementations. This will bring different programming libraries and thus a welcome diversity. But there are shared libraries that are used by different programming languages. Their position is so dominant, they are very hard to replace with an independent implementation. E.g. OpenSSL, GMP etc. Special care has to be taken to guarantee diversity on this level.

Modern CPU level exploits like Spectre / Meltdown etc. can be avoided by working with dedicated hardware for the control components that are installed directly on the physical server. However, the hardware servers will still have an onboard management interface that serves as a backdoor into the system for setup and maintenance of the servers. Special care needs to be taken to close this access. But experience shows that you can not be really sure you closed every backdoor in this regard. So beyond hardening, there also has to be diversity of the hardware, including CPU, mainboard, various controllers with their firmware etc. in order to mitigate the threat that comes form these management interfaces or supply chain attacks.

The Verifier

With all this being said, it goes without saying, that diversity should also stretch to the verifier. There are four control components and there has to be a strong requirement to have multiple and diverse verification software to raise the cost of a successful attack.

Summary

A high degree of diversity is a core requirement for a trustworthy internet voting system. The priority should be with the application software of the control components since an attacker is expected to attack this software first. But since all the 3rd party components (Shared libraries, OS, firmware, hardware, network devices etc.) are cheaper to diversify, diversification should extend to these parts of the system as well.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation either in writing or a simple upvote. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT: Typos]

[EDIT: Introduced the keyword "supply chain attack"; rephrased trust by the public bc of hardening audits; rephrased description of increased attack surface with multiple control components; reworded description of attacker's preferences (Appl vs OS).]

[EDIT: Introduced the keyword "compiler" in the list of software components]

[EDIT: Added formal verification methods, replaced "remedy" with "relief", added paragraph about conformance testing btw different implementations and rephrased the paragraph about the operational costs. All based on @David.Basin 's input below.]

[EDIT: Reworded the exponential costs of conventional hardening and mentioned diminishing returns based on @Bryan.Ford 's proposal below. Put OpenSSL in front of GMP after Bryan pointed out in 2B, there are in fact alternatives for GMP.]

[EDIT: Tuned down the importance of the audits for public trust, added a sentence about operational separation of the control components and runed down the wording around the relative costs of attacking the OS vs application software; all based on @Florian.Egloff 's input below.]

Edited by Christian Folini 6 months ago

Christian Folini @christian.folini added Last-Call label 6 months ago

Christian Folini @christian.folini mentioned in issue #12 (closed) 6 months ago

Christian Folini @christian.folini mentioned in issue #13 (closed) 6 months ago

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

Just spotted a typo in: "It seems liely that the application software"

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Olivier. Fixed.

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

Thanks for the great summary! I fully agree that diversity is a most effective way of limiting the risks of attack strategies that would have been overlooked, or that rely on elements that are largely out of the control of the system provider (external libraries, OS, hardware, network equipment, ...). There are also little benefits in making a distributed system architecture (multiple control components, ...) in which we rely on at least one component being honest, if any attack is likely to succeed equally on all the components, because these components are just the same.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

Following up on my comment in 2C. I am not in the know to what extent supply-chain attacks are currently a threat or not, but I am sure that they could be true. I am happy with your summary. It is good and covers supply-chain attacks in spirit (by increasing the likelihood of detection through diversity).

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Carsten. I have added the keyword "supply chain attacks" to make sure it is there in writing - and not only in spirit. :)

Florian Egloff @Florian.Egloff · 6 months ago Developer

I agree with the diversity claims overall, but have some comments on the specifics (I am using direct quotes from your statement for clarity):

> There still needs to be a high level of hardening and scrutiny, not the least because the perception of the public will depend a lot on these audits.

Instead of "perception of the public", I would rephrase to something like "because public trust will depend...". you are correct that it is the public perception that depends on it, but as it is written, you risk adversarial interpretation of doing "public perception" management.

> There is also a bigger attack surface with multiple systems. But these additional exploit options are limited to individual control components and they no longer harm the security of the overall system.

This, of course, depends on the attacker having no benefit from having exploited one control component to attack the other control components. I would make this explicit. If this is not the case, the larger attack surface really does matter.

> So in consequence, the OS and other 3rd party components will only be attacked if it is cheaper than attacking all four control components together.

I am not sure one can make this blanket statement. A (state) attacker's operations will have a different approach to exploit management. They will have an up-front investment in being able to attack shared platforms, which will then be discounted over the period of use. Thus, you are still correct with the "cheaper" judgement, but only if viewed from an operational perspective.

In other words: In absolute terms, the exploit development for the application layer may be cheaper, whilst for the attacker in relative terms (already possessing exploits for OSs), it may be more expensive. I have no good suggestion of how to fix the statement, however.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Florian. These are valuable remarks.

1. Reworded based on your proposal.
2. Made clear this is only true if the control components are truly independent.
3. Replaced "cheaper" with "brings an advantage".

Could you please check if this is now clearer / more to the point?

Florian Egloff @Florian.Egloff · 6 months ago Developer

Sorry for the delayed feedback.

1. It reads better, though I would be less sure about the judgement you inserted. You wrote, "public trust will depend a lot on these audits". The "a lot" assumes we know how much audits matter for public trust. I would, in the absence of evidence, just drop the "a lot".

2. You wrote: "But the more independent the control components are, the less can an attacker profit from one exploit when attacking the next one." This does not quite address what I meant. My claim here was less about the independence of the stack, but more about the independence of access. The thought was that if we have a larger attack surface, due to diversity of components, then the attacker can choose which element to compromise first. The compromise of one component should then NOT give the attacker an advantage over compromising the other three (e.g. using the compromised control component as a bridge for the attack). This is largely a function of operations & segmentation, and NOT a function of independence of the control components. We can only "neglect" the security downsides of expanding the attack surface if this segmentation holds. In any case, it should be part of the risk assessment.

3. You write: "However, exploits for widely used software and hardware components are very expensive to obtain and using them in an attack where they could be discovered makes it very unattractive to use them. So in consequence, the OS and other 3rd party components will only be attacked if it brings an advantage over attacking all four control components together." This is outside of my depth of expertise, and in my opinion, we should really have someone from the offensive operations side weighing in on this. Here is what I think, but again, ask an offensive operations specialist: For me, "expensive" is relative to the actor you are (if you are a state, it may matter less, for reference see https://zerodium.com/program.html). The "unattractive to use" is a function of defensive security practices employed (example: who was able to spot EternalBlue before it was publicly released?). And the conclusion that OS/3rd party components will only be attacked if they are more advantageous than the control components rings true but also does not bring us in a very strong analytical space, where we would have traction over the question "how hard is hard" here (i.e. how much of a difference does shifting the attacker's focus towards attacking the OS actually make?).

I hope these comments are helpful.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for checking. I'm updating the text again.

1. OK.

2. I see. I misunderstood you the first time. I'm adding a sentence about operation and segregation.

3. I agree with you that guessing the incentives of potential attackers is very hard and the whole paragraph saw very few comments from the experts. Not surprising, this is an area where we lack hard data. A different set of experts might have given different answers, but we not simply expand it at this stage. But what we can do, is tuning down the wording a bit, so it no longer sounds as if we had said hard data.

Now: "... makes it very unattractive to use them. So in consequence, ..." Update: "... makes it unattractive to use them. So, as a

potential consequence, ..."

Florian Egloff @Florian.Egloff · 6 months ago Developer

Perhaps some more operationally experienced from the @federal-office section could comment on (3)? @Gerald.Vernez @Riccardo.Sibilia @Reto.Inversini

Christian Folini @christian.folini mentioned in issue #28 (closed) 6 months ago

David Basin @David.Basin · 6 months ago Developer

I am in general agreement with summary but there are some specifics I might formulate differently.

1. "while more hardening and more scrutiny will bring exponential costs, but less and less gain in terms of security." Yes for hardening and scrutiny. If we consider formal verification, then the situation is different: I can have 100% guarantees FOR WHAT I VERIFIED. If I verify a program (say a database system) I can have exact guarantees about its behavior. But these guarantees assume that the OS, hardware, and any library functions I didn't verify meet their requirements. Please let's not lump all quality control measures together and forget that programs and systems can be treated as mathematical objects and subjected to exact analysis methods.

2. "Thompson came to the conclusion, that "No amount of source-level verification or scrutiny will protect you from using untrusted code." Diversity is one remedy for this problem, namely for Swiss internet voting. This is because it forces an attacker to exploit all four control components in parallel. This is (a) very expensive for the attacker and (b) could lead to the attack being detected, which is very risky for an attacker."

Yes, diversity makes the problem harder (as the last 2 sentences above indicate). No it doesn't remedy the problem. That said, we are unlikely to remedy the problem Thompson mentions, and it highlights that we always must be clear about the assumptions behind the proper functioning of our systems.

3. "This does not mean, that hardening and scrutiny are futile. There still needs to be a high level of hardening and scrutiny, not the least because public trust will depend a lot on these audits."

I would prefer to distinguish quality assurances measures like hardening and scrutiny that are best effort without exact guarantees ("well, we did our best and we think it works") from mathematical/logical methods that can provide guarantees. This may seem pedantic, but the distinction is important. Too often activities like testing, verification, audit, pen-testing, ... are lumped together but the kinds of guarantees they provide are very different.

4. "At first sight it also includes a better specification, since independent developers will need to be able to write their code based on the specification alone. A single implementation can be developed based on a specification that has some ambiguity. But with a second, independent implementation, the specification is being validated by 3rd party developers, the ambiguities will surface and will have to be cleared out."

Unclear. Only if some kind of conformance testing of the two implementations is done.

4. "Operation is also more expensive with multiple independent systems. However, it is fairly obvious, that the costs are close to linear,"

Probably depends if functionality is to be implemented to reconcile and recover from differences.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you very much David. I have tried to address your concerns one by one.

1. "While more hardening ..."

This is an additional aspect. I have introduced the formal verification method into the section under the title "Hardening and Scrutiny Still Important". I hope the argumentation did not get too complicated this way.

Please check.

2. "Thompson came to the conclusion, ..."

Good point. Maybe it's my German mother tongue where the translation does not imply a complete solution as much as "one remedy" does. I have replaced it with "a relief". Better?

3. "This does not mean that ..."

The item number 1 that you raised was addressed by extending this paragraph. For what is worth, I think the two items are very close and I think it is OK to address them together?

Is that OK with you?

4. "At first sight ..."

I have added a new paragraph about conformance testing.

5. "Operation is also more expensive ..."

I am not entirely sure I would add these potential costs to the operational costs of the application software itself, but I see there are reasons to do so. I do not want to drag this discussion out, though, so I have added the reconciliation problem to this paragraph while still maintaining that the costs are close to linear. I think this is in the spirit of the discussion here and if we drop it, it could be used to skip the whole diversity idea because of potential operational costs.

What do you think?

Bryan Ford @Bryan.Ford · 6 months ago Developer

Christian, fantastic summary in general.

Like David, I also found this particular phrase confusing and perhaps problematic:

> more hardening and more scrutiny will bring exponential costs

What exactly does this mean, and in particular what does "exponential" mean in this context? It's not clear to me that more hardening and/or scrutiny would bring "exponential" costs... or exponential with respect to what? Financial investment? That doesn't seem right: presumably the cost and scrutiny that you get by paying for it would be approximately linear in the person-power you hire, no? Or do you mean it'll take exponentially more effort to find each successively more subtle flaw or bug after you've already found most of them? That seems at least a bit more plausible, but I'm not sure that's what you're trying to say and I don't know whether that's true.

Or did you perhaps mean something more like, "...while more hardening and more scrutiny will bring less and less security gain as investment increases"? Perhaps it's worth mentioning that the "law of diminishing returns" applies in this context, in that it gets harder and more costly to find and fix the next progressively more-subtle flaw as all the easy/obvious bugs get shaken out?

And I think David is right that formal verification is a somewhat different beast from ordinary hardening and scrutiny, and probably requires a different kind of cost/benefit analysis, although I'm not sure to what extent or in what way that should be addressed in this summary. With conventional hardening/scrutiny, you get a lot of benefit early on at low cost because you catch all the easy bugs, but then the remaining bugs start to get harder to find and you get into the diminishing-returns problem to find the rest. With formal verification, you pay a pretty huge cost up-front to produce the formal specification and formal proof of the implementation, but also with the correspondingly huge payoff that you've guaranteed that there is a 0% probability that any bugs remain in what you've actually verified, modulo flaws in the specification and/or verification tools. I'm not sure that even that up-front verification cost should be described as "exponential" (in what?) - just a hefty but roughly-constant multiple of standard development costs sounds more likely to me, but David's the expert on that. But again, I don't know how much the summary should get into the cost-benefit analysis distinctions of conventional hardening/scrutiny versus formal verification.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Bryan. Your reference to the Law of Diminishing Returns or maybe even Brooks's law is exactly what I had in mind but failed to express clearly.

I have replaced my wording with your proposal and added a sentence about Diminishing Returns. I did not use the term *law*, since Wikipedia does not list it as law, but it has a lemma Diminishing returns.

David Basin @David.Basin · 6 months ago Developer

I am fine with "diminishing returns" rather than trying to characterize the nature of the returns with some particular (exponential) function.

More as an aside, I wanted to point out that finding bugs in large code bases is easy (https://dl.acm.org/doi/10.1145/1052883.1052895) and one needs to be careful in updating static analyzers as churn (finding too many new bugs) tends to upset the customer (https://dl.acm.org/doi/pdf/10.1145/1646353.1646374)

Said another way, if you take a large code base developed without formal proofs, and run a(n upgraded) static analyzer on it,

experience suggests you are likely to find bugs. This is, of course, particularly relevant for critical systems like voting.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you David.

Interesting input on the question of finding bugs. We'll turn to that problem soon when introducing risk assessment into the discussion and the question how to treat relevant bugs in production e-voting code.

Christian Folini @christian.folini mentioned in issue #29 (closed) 6 months ago

Christian Folini @christian.folini mentioned in issue #27 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 2B - Independent Operating Systems for Control Components used for return-code generation (Block 2 - Diversity)

Reference to originating discussion block

Block 2 - Diversity

Question

From the answers to question 2.1.6 in the questionnaire we take that experts tend to advise using different operating systems for the critical components. By relating to the above question, we would like to get a better understanding of the added value and the downsides.

Which puts the correct functioning of at least one control component to a greater risk: Using the same operating system or using the same application layer software on all four control components?

Edited 6 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to January 02, 2021 7 months ago

Christian Folini @christian.folini added Block-2 Cryptography labels 7 months ago

Christian Folini @christian.folini changed due date to February 02, 2021 7 months ago

Christian Folini @christian.folini mentioned in issue #10 (closed) 7 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

There has not been any comment on this question so far. Maybe the question is too close to A - or the question is too painful as it is posed in a way that forces you to make a call in a dilemmatic situation.

The point is, the cantons and the system provider are facing this (and many similar) dilemmas: Resources are finite and they need to set the priorities somehow.

Almost all experts who responded to question 2.1.6 made it clear that independence should go beyond the application software stretching to the operating system, standard libraries, firmware and hardware.

@Eric.Dubuis, @Rolf.Haenni, @Reto.Koenig and @Philipp.Locher from BFH have taken a different take for 2.1.6 however. If I am reading it correctly, BFH thinks total independence is unrealistic, so it's better to make sure to run secure code. Maybe that is a more realistic and entirely pragmatic approach.

Does this convince you?

Now back to the dilemma: If you experts don't provide guidance here, then somebody else will have to make the call. But if you do provide guidance, and you speak with a clear and strong voice, then it will be hard to brush it away.

So I'm inviting you to think about this question and give us your opinion.

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

I think it's a question of (a) whom you consider feasible attackers to be and (b) which parts of your process you can verify independently. If you consider Microsoft to be within your attacker model, then you need to have at least one control component that they don't control (unless it suffices to verify that component's computations, which at least in the current protocol it doesn't). If you don't consider Microsoft to be a potential adversary, then it may not matter that all the control components are running Windows, assuming it is patched. If every control component is running the same up-to-date operating system, then your election

can be compromised by a non-Microsoft party that finds a 0day attack. This is a problem if every other method of subverting the election is harder than finding (or more expensive than buying) a 0day attack against that operating system...

(and of course for 'Microsoft' read also 'Android/Google', 'Apple', etc.)

So I don't think there's a debate here, so much as a decision about whether these sorts of attacks are in scope and whether the rest of the system should be assumed to be hardened against attacks of similar cost.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for picking this up, @Vanessa.Teague: Your statement is well to the point.

If I align it to the very exact question, I get the following:

It is more important to run different application software on the control components. The operation system will only be attacked when a 0-day against the operation system is cheaper to buy or develop than exploiting the application software on the four control components.

Am I understanding you correctly?

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

Well, we can apply the same reasoning as before. If we have at least four different OSs, and we assess the probability of a catastrophic exploit being found in any of them at any time as the value p, then if all components run the same OS, then the risk of all of them being susceptible to a zero-day attack at the same time is p. If they all run different OSs, then the chance of them all being open to failure on day zero is p^4. Isn't that right? I don't see a difference in whether you are considering an OS or another layer of software support.

Edited by Oscar Nierstrasz 6 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

We should consider that if you put state adversaries into scope, the likelihood of having at least one 0day exploit for the OS's becomes very high. After all, these are common platforms across many of their target sets, where it is worth to spend money and effort on finding vulnerabilities you can use. It would be imprudent to assume that all the money spent on offensive cyber capabilities would not result in some capability against the major OSs.

Having said that, and as @Bryan.Ford elaborated beautifully in #10 (closed), and @David-Olivier.Jaquet-Chiffelle laid out very clearly in his answers in the questionnaire (e.g. already in 1.1.), independence also mitigates unintentional failures. Thus, both, having independence in the application layer and OS are important.

Christian Folini @christian.folini changed title from **Discussion B - Independent Operating Systems for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 1B - Independent Operating Systems for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Christian Folini @christian.folini changed title from **Discussion 1B - Independent Operating Systems for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 2B - Independent Operating Systems for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Tobias Ellenberger @Tobias.Ellenberger · 6 months ago Developer

The attack scenarios are important to be able to give advice. Different combinations are possible:

1. same OS same application
2. same OS different applications
3. different OS, same application
4. at least two different OS, same application
5. at least two different OS, different applications
6. all different OS, same application
7. all different OS, different applications the list is not complete.

It can be assumed that 0-days are or will be available for every OS or application. Not only from the aspect already mentioned several times also in other threads (see Post Florian Egloff above in this thread), but also because the complexity increases not only in hardening and operation of the OS and Applications, but also in the execution of the attack itself. Thus, with increasing diversity and complexity the chance that the attacker will make mistakes and the attack will be detected increases.

Edited by Tobias Ellenberger 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Tobias.

Are you referring to this post by @Florian.Egloff?

And am I correct when I read your response as "The more diversity on all levels the better?".

And - answering to the question more directly - what is the priority for additional funds? Independent additional control component or supporting an independent additional operating system?

Collapse replies

Tobias Ellenberger @Tobias.Ellenberger · 6 months ago Developer

> Are you referring to this post by @Florian.Egloff?

to the one in this thread

> And am I correct when I read your response as "The more diversity on all levels the better?"

if there are "unlimited ressources" both financially as well as manpower and knowledge: yes.

> And - answering to the question more directly - what is the priority for additional funds? Independent additional control component or supporting an independent additional operating system?

mostly it is the other way round isnt' it? you have a fix amount of (additional) funds and then you have to deal with it. But if you have unlimited budget or the budget is made based on our needs, i would recommend to go for an additional control component.

Christian Folini @christian.folini · 6 months ago Maintainer

Ah, thank you. I was not sure about the thread. And everything clear with your response now.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 6 months ago Developer

Creating a trustworthy eVoting system is much more demanding than just an eVoting system, and some trade-offs are not acceptable in order to get close to a trustworthy system. It would be wrong to start thinking in term of “how to optimally use a fix amount of (additional) funds”. This does not mean that we have to suppose unlimited funds or manpower. Even if there are limited resources and manpower, there are core requirements that cannot be circumvent. If not enough funds are available to reach these core requirements, then the (aimed to be trustworthy) eVoting system should not be used, not even in a large-scale test version, until extra funds are granted and these requirements are fulfilled in a satisfactory way. I consider that redundancy at all levels is one of the core requirement for a trustworthy eVoting system. I deem it necessary (although not sufficient). As for any sensitive physical system (nuclear plant, aeronautic, critical infrastructure, etc.), redundancy should be omnipresent in an Internet Voting system. Redundancy allows to discover some possible malfunctioning. Sometimes, redundancy is necessary for cross-verification. It makes many attacks much harder to achieve and often easier to detect. Moreover, redundancy should be built on independent components at all levels (physical components, OS, protocols, programming language, application layer, printing office, etc.) From my perspective, more generally, trustworthiness requires four conditions to be fulfilled simultaneously with no compromise: (i) Everything reasonable (omnipresent redundancy at all levels, independent systems, use of well-studied crypto and protocols, etc.) needs to be done to avoid problems and have to be regularly verified by trusted, independent entities (ii) It is recognized that problems are expected to happen anyway (iii) Any problem happening is highly likely to be detected (iv) Any detected problem is highly likely to be solved in an appropriate way

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you for this constructive contribution David-Olivier. May I reformat it a bit for better readability?

Christian Folini @christian.folini · 6 months ago Maintainer

@David-Olivier.Jaquet-Chiffelle, we have heard many different opinions on the need for independence on the control components. Your statement is the most radical though. That's why I would like to make sure we understood you correctly.

You are certainly right that there is a minimal set of requirements that have to be fulfilled in order to qualify for a future online voting

system. What we are discussing here is the extent of these minimal requirements, possibly with a risk-based approach in mind. The way the question was asked is meant to give the regulator an idea which requirements are more important than others (Independent implementation of control component? Independent OS? Independent hardware?). Which attacks are more likely? Which single point of failures are more risky than the other ones? Your statement underlines the need for independence on all levels, which is a radical, but fair point in my eyes.

I think the central part of your statement is this:

Moreover, redundancy should be built on independent components at all levels (physical components, OS, protocols, programming language, application layer, printing office, etc.)

"Independent protocols" is a new aspect, I think, since I have not seen this with the other statements. Could you elaborate this please? Is this the protocol used for the encryption / mixing / decrypting of the vote?

Furthermore, we became aware that the GMP library is so dominant, there seems to be no alternative to this library for the crunching of big numbers. So is this a killer as it seems unlikely to get an independent replacement here?

And finally, independence is very difficult in the printing process due its physical nature. That's why block 3 brings two propositions that try to get rid of the trust assumptions about the print office. I hope that addresses some of your concerns.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

I think this is to be read as "independent implementations of the protocol".

Christian Folini @christian.folini · 6 months ago Maintainer

I think so too. Thanks Carsten.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 6 months ago Developer

Yes, for situations with no relevant alternative for the protocol or the algorithm, independent implementation in different programming languages (and without using the same core/external libraries) is important to create enough independence. The implementations should be developed by different and independent (private and/or public) companies. The companies should not be owned by common entities (no common financial or political agenda) and should not share or use common resources (in particular human resources, or the same standard libraries). Then, each program should be compiled with two independent compilers (a single compiler should not be trusted, this is a central point) and run in parallel on different servers (brand, processors) with different operating systems.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you @David-Olivier.Jaquet-Chiffelle . I agree with this. I have enhanced the summary in 2A and added the compiler into the list of software components that have to be diversified.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I think the latest summary in 2A does a great job of capturing these important points.

At the risk of a minor diversion, though, I'd like to push back a bit on this statement:

> the GMP library is so dominant, there seems to be no alternative to this library for the crunching of big numbers

I'm not sure where this came from, but I actually don't agree. Even in C-language land, there are at least two different, widely-used and reasonably well-maintained big-number-arithmetic libraries that I know of: not only GMP but also the BIGNUM library in OpenSSL. The latter isn't as feature-rich but it's designed with crypto in mind. Unless I'm mistaken, those libraries are largely if not fully independently maintained (I think they even use incompatible open source licenses, which likely prevents cross-fertilization of code in at least one direction).

In the Go language, which my group uses a lot, there are big-number arithmetic libraries built into the standard library (big.Int). I'm fairly familiar with this library, as my group's Kyber library for advanced crypto uses it extensively, and at one point I put some effort into evaluating what it would take to make it guarantee constant-time operation for crypto uses. I can attest that there definitely doesn't seem any dependency on GMP or any other external library there: it's implemented in pure Go with architecture-specific assembly optimizations. Some of it may originally have been derived or inspired from developers looking at the GMP and/or OpenSSL BIGNUM libraries, but it's certainly independently maintained now, got thoroughly rewritten and redesigned in the translation, etc., so I'm pretty confident in its independence at this point.

I know that the Rust language now also has big-number support, though I'm less familiar with its implementation or maturity level. I don't know offhand but would be extremely surprised if its implementation depended on GMP.

Finally, even JavaScript is now getting language-level BigInt support. I know even less about how it's implemented in the major browsers and other JavaScript virtual machines - in this case I find it more plausible that the implementations might just link in GMP or BIGNUM underneath. But again I don't know.

At any rate, in summary I'm pretty confident that there are at least four different big-number arithmetic libraries available in at least

three different languages, I'm pretty sure they're reasonably independent if not fully independent, and there may well be others I'm just not aware of. So I don't think the GMP dependency situation is so bad.

What *is* kinda bad is that practically none of those libraries that I know of - except perhaps for OpenSSL BIGNUM - seem to have put any significant effort into supporting constant-time operation for crypto usage. But that's another matter.

Rolf Haenni @Rolf.Haenni · 6 months ago Developer

Dear Bryan

GMP allows constant-time modular exponentiation using

- gmp_lib.mpz_powm_sec

The documentation of this function says: "*This function is designed to take the same time and have the same cache access patterns for any two same-size arguments, assuming that function arguments are placed at the same position and that the machine state is identical upon function entry. This function is intended for cryptographic purposes, where resilience to side-channel attacks is desired.*"

But as you said, this is another matter.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you very much for adding this perspective Bryan. Very welcome.

The potential problem with GMP was first mentioned in the BFH response to 2.1.6.

I have picked this up and asked for more infos in 2C, but did not get a response. So I took it into the summary with the infos I had.

You mention several alternatives that seem viable. Thanks.

The summary that mentions GMP lists it as an example area where special care needs to be taken to guarantee independence. I think the need for special care persists and if we remove GMP from the list only OpenSSL remains. So I'd prefer to tune it down a bit. OpenSSL is very dominant for TLS obviously, but alternatives exist. From what I learned about GMP, the situation is fairly similar. So we could for example OpenSSL in the first position. Or we tune it down even more by adding "to a certain extent" to the GMP or something along those lines. What do you think?

BTW, I also tried to contact the GMP developers via their discussion mailinglist. I did not get a response though, but a private message from an anonymous person claiming to work in aerospace pointed to the fact that the lead maintainer would not want to discuss security of GMP in public. I did not pursue this further.

Bryan Ford @Bryan.Ford · 6 months ago Developer

@Rolf.Haenni it's good to hear GMP has at least one important constant-time operation. But that's just one of quite a few that really should have constant-time implementations in order to ensure leak-freedom convincingly.

@christian.folini given that OpenSSL/BIGNUM is a big-number library *for crypto*, I would indeed put it in first place. I don't know much about the politics behind GMP, bit it sounds to me like the main developers/maintainers never designed it with crypto in mind and don't want to take on the responsibility or any (real or perceived) risks related to use of GMP for crypto. Unfortunate but understandable. So even though there is a reasonable diversity of big-number libraries across multiple languages - so it's fairly easy to get big-number-library diversity as a side-effect of multi-language diversity - the problem really becomes that none of these big-number libraries *except* OpenSSL's is actually designed for or really "wanting" to support crypto use-cases securely.

As it happens, as a first tentative and experimental step to try to remedy this situation, my lab recently launched a student project to prototype basic constant-time crypto libraries across multiple languages. There's no expectation that code coming out of a student project like this would be remotely something we want to depend on for real, of course. But the hope is that if we can take a first step in a moderately diverse cross-language fashion, others - perhaps first at other academic institutions and later in industry - may get interested and become willing to get involved to develop, mature, harden, and maintain such a cross-language crypto-focused suite. (If any of the other experts here or your students/colleagues might be interested in collaborating with us on this, get in touch!) But we'll see - it's only a first tentative step, and it certainly doesn't change the fact that the current situation in crypto-hardened arithmetic libraries is pretty grim.

Anyway, apologies again if this is a technical diversion away from the more critical big-picture issues we're trying to discuss here...

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you very much Bryan. We really appreciate your input and if it helps us gain a wider perspective, then it's double welcome. Please keep those ideas coming.

Christian Folini @christian.folini · 6 months ago Maintainer

The discussions 2A, 2B, 2C and 2D are summed up together in the summary for 2A.

Christian Folini @christian.folini added Last-Call label 6 months ago

Christian Folini @christian.folini mentioned in issue #12 (closed) 6 months ago

Christian Folini @christian.folini mentioned in issue #13 (closed) 6 months ago

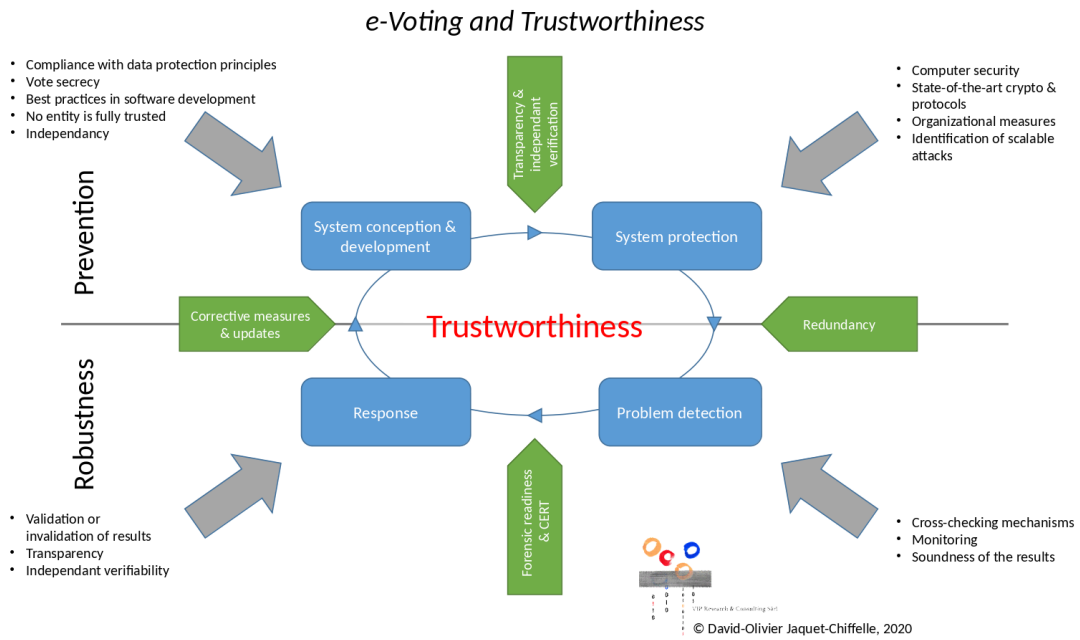
David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 6 months ago Developer

I think that one of our challenges in the expert discussion (and for the report that will follow) is to keep in mind the global picture and, in particular, the interdependancy of the different conditions to fulfill eVoting_map.pptx in order to reach a trustworthy eVoting system. I created a map that summarizes these conditions and their interdependancy. I decided to share it with you. I believe that it can help situate each specialized discussion within the global picture and make sure that each condition is given enough importance and is eventually covered with a similar depth. It is a first attempt and you might have constructive suggestions to improve it.

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you very much @David-Olivier.Jaquet-Chiffelle.

Here is a png version of the pptx file:



Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary in 2A as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 2C - Independent CPU for Control Components used for return-code generation (Block 2 - Diversity)

Reference to originating discussion block

Block 2 - Diversity

Question

Same as question B for different hardware.

Given physical servers for the critical components, which puts the correct functioning of at least one control component to a greater risk: Using the same operating system or using the same type of CPU in all four control components? Are other hardware components more critical than the CPU (find a list of proposed components in the summary to question 2.1.6 of the questionnaire)?

Edited 6 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Block-2 Cryptography labels 7 months ago

Christian Folini @christian.folini changed due date to January 03, 2021 7 months ago

Christian Folini @christian.folini changed due date to February 03, 2021 7 months ago

Christian Folini @christian.folini mentioned in issue #10 (closed) 7 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

Actually, I am curious myself, if a different processor could help mitigate some of the risks that comes with using the GMP library for computing large numbers. GMP is really dominant in this domain after all.

@Ulrich.Ultes-Nitsche called for using more than one type of CPU, but nobody offered a realistic alternative to GMP so far. So I wonder if a big-endian processor would force an attacker to exploit GMP twice. Or maybe that's just naive thinking of somebody who never wrote assembly language.

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

My two cent opinion: AFAIK exploits are far more common in OSs than in CPUs, so the risk should be higher in adopting the same OS for all components, than in running them all on the same hardware. The question is, what is the probability p that an exploit will be discovered for a given OS or a given CPU?

Adrian Perrig @Adrian.Perrig · 6 months ago Developer

CPU bugs can be used obtain information about other processes running on the system (by exploiting hardware side-channel attacks). It all started with Spectre and Meltdown, now there's an entire zoo of exploits, for instance this paper that just received the best paper award at the Oakland conference yesterday: https://download.vusec.net/papers/trrespass_sp20.pdf Although several CPUs are vulnerable to such attacks, diverse HW can nevertheless lead to improved assurance (assuming systems are used that can tolerate compromise of a subset of entities).

A hardware component that is used on many systems is a management engine / active management technology, for instance the Intel management engine. This technology is embedded in numerous systems (mainly servers), enabling remote management. The problem is that they are running on a separate CPU, have a separate network interface, and are typically running large code bases. In essence, it's a completely independent system that can be compromised and it can then control the main CPU. It is hard to find

systems with secure connections, some of these systems for instance still use SSLv2 to secure the management connection!

Christian Folini @christian.folini changed title from **Discussion C - Independent CPU for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 1C - Independent CPU for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Christian Folini @christian.folini changed title from **Discussion 1C - Independent CPU for Control Components used for return-code generation (Block 2 - Diversity)** to **Discussion 2C - Independent CPU for Control Components used for return-code generation (Block 2 - Diversity)** 6 months ago

Philippe Oechslin @philippe.oechslin · 6 months ago Maintainer

Although hardware attacks like Tresspass, Specter et al look scary, they apply to a specific threat model. The threat is malicious code that uses a hardware side channel to extract data from other processes. The code is typically run by a malicious tenant in a multi tenant setting (cloud), or served by a malicious web site to be executed in the victims browser.

In the case of the control components we have bare-metal machines executing only the OS and the software needed for executing the protocol. It is a single tenant system and it doesn't download any code.

The only case I would see where this side channel could be used is if a non privileged process (e.g. a health monitor) tries to access secret information of a process holding secrets.

To exploit such a vulnerability, an attacker would have to install malicious code on the control component without being detected.

The VEleS (Art 7a) mandates that the source code of all software running on the control component must be published unless the software is widely used and regularly updated (e.g. OS, database, web server, ..).

Even if these attacks can be devastating in other settings, it seems that the risk of a processor or memory bug being exploitable to successfully extract data from a control component is very small compared to other risks.

Am I missing something?

I am more concerned about undocumented or buggy management engines. Diversity of processors would certainly reduce this type of risks.

Edit: s/wildly/widely

Edited by Philippe Oechslin 6 months ago

Tobias Ellenberger @Tobias.Ellenberger · 6 months ago Developer

> Even if these attacks can be devastating in other settings, it seems that the risk of a processor or memory bug being exploitable to successfully extract data from a control component is very small compared to other risks.

agree with that.

While reading it, i asked myself the question: the tendency of the recommendations in the other discussions is towards different OS, different applications. Can't different hardware be recommended for the same reasons, considering that we don't know which will be the next "similar" hw-channel-attack like Meltdown/Spectre? Think the cost <-> potential benefit ratio is relatively small.

Christian Folini @christian.folini mentioned in issue #11 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

The discussions 2A, 2B, 2C and 2D are summed up together in the summary for 2A.

Christian Folini @christian.folini added Last-Call label 6 months ago

Christian Folini @christian.folini mentioned in issue #13 (closed) 6 months ago

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

@Adrian.Perrig and @philippe.oechslin, I agree with both of your statements. I want to add that there is still the possibility of a supply chain attack, in case dedicated hardware is used. See, for example, https://www.bloomberg.com/news/articles/2018-10-04/the-big-hack-the-software-side-of-china-s-supply-chain-attack

Christian Folini @christian.folini · 6 months ago Maintainer

Thanks for bringing supply chain attacks up. I have already written the summary and I did not include this keyword. Do you think it should be added, or is it there "in spirit"?

Bloomberg: I think you are totally right with stipulating that China could easily perform such an attack if they wanted to. But most industry experts did not believe this particular story. In the end it does not matter too much, since we agree it could be true and we need to defend against it.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I generally agree with everything in this thread, and think the summary is good. Especially the extremely important points about management engines and supply-chain vulnerabilities; glad you got those in explicitly now.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Bryan.

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary in 2A as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Discussion 2D - Control Components used for mixing and verifier (Block 2 - Diversity)

Reference to originating discussion block

Block 2 - Diversity

Question

The previous questions are about the control components used for return-code generation. Here we would like to find out if your assessment is different when considering the other control components and the verifiers. (We will revisit the printing office in subsequent questions.)

Is your answer to questions A, B or C different when considering

- *the control components used for mixing?*
- *the verifiers?*

Edited 6 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Block-2 Cryptography labels 7 months ago

Christian Folini @christian.folini changed due date to January 04, 2021 7 months ago

Christian Folini @christian.folini changed due date to February 04, 2021 7 months ago

Christian Folini @christian.folini mentioned in issue #10 (closed) 7 months ago

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

An important difference is that the control components used for mixing (and decryption) are trusted for privacy, but are not supposed to be trusted for verification. That is, even if an attacker compromises all four of them, it is still not meant to be possible to manipulate votes.

The verifiers are a different thing - I would hope that verification could be much broader than it currently is in Switzerland.

Christian Folini @christian.folini changed title from **Discussion D - Control Components used for mixing and verifier (Block 2 - Diversity)** to **Discussion 1D - Control Components used for mixing and verifier (Block 2 - Diversity)** 6 months ago

Christian Folini @christian.folini changed title from **Discussion 1D - Control Components used for mixing and verifier (Block 2 - Diversity)** to **Discussion 2D - Control Components used for mixing and verifier (Block 2 - Diversity)** 6 months ago

Christian Folini @christian.folini mentioned in issue #11 (closed) 6 months ago

Christian Folini @christian.folini mentioned in issue #12 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

The discussions 2A, 2B, 2C and 2D are summed up together in the summary for 2A.

Christian Folini @christian.folini added Last-Call label 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my updated summary in 2A as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 6 months ago

Christian Folini @christian.folini removed Last-Call label 6 months ago

3-print-office.md 11.5 KB

Discussion Block 3 - Print-Office (Diversity to support security and trust-building - Part 2)

Generation, verification and printing of parameters and codes

In their answers to question 2.1.3, mandated experts expressed that trusting the printing office at performing critical computations without verification is problematic. A majority seems to consider it highly problematic. Computations should be avoided or verified.

At some point, the task-force will recommend which measures to implement and by when. To that end, we make two propositions which we ask you to assess. The first one would take time for cantons and providers to implement. The second one is simpler but brings less value. It is meant as a possible mitigation to allow more time for the first proposal to be implemented.

Proposition A

1 - The notion of the printing service is replaced by “setup-components”, “setup-verifiers” and “printing components”.

2 - Functionality of the components

2.1 - Setup-components

- They generate the security-relevant parameters as defined by the cryptographic protocol, including the codes for individual verifiability.
- They use multiple random sources for generating random values. As an alternative for public parameters they could use a seed agreed upon by multiple persons.

2.2 - Setup verifiers

- They verify that the set-up components have generated the parameters correctly.

2.3 - Printing components (machines that receive the verified file containing the parameters to be printed, as well as the printing machine)

- They verify the origin of the verified file containing the parameters to be printed (signatures of setup-component and setup-verifier).
- They decrypt the file.
- If the file is not yet in a printable format, they create a printable file.
- They print the voting cards.

3 - Trust-assumptions (assuming one setup-component, one setup-verifier and one printing component)

3.1 - It is trusted that either the setup component generates the parameters correctly or the setup-verifier verifies the correct generation soundly.

3.2 - One out of four random sources is trusted to deliver values with sufficient entropy (control-component or human chosen secret passphrase/dice roll).

3.3 - With regard to verifying secret parameters generated based on values established by persons: It may be assumed that persons will not be able to memorize very long values, e.g. the hash of the other persons’ passphrase.

3.4 - The setup component or the setup-verifier is trusted not to leak secret data, unless by encoding data into their output.

3.5 - The printing component is trusted:

- To verify signatures soundly
- To print according to the verified printing file
- Not to leak secret data

3.6 - The channel between the setup-component and the setup-verifier is trusted. Channels leaving these components are untrusted.

4 - Operational requirements for “setup-components”, “setup-verifiers” and “printing components”

4.1 - Before performing critical operations the following measures must be taken:

- Components have to be secured prior to operation
- Network connections must be removed physically

- The origin of the software to be installed has to be verified. The origin of the internet voting software has to be verified based on comparison with a signed published representation

4.2 - After performing critical operations the following measures must be taken:

- Components and any physical support holding critical data must be securely stored after use or the data securely erased
- Unless there is a significant reason, data is securely erased after use.

4.3 - Organizational measures need to enforce the consent and presence of a sufficient number of persons in the roles necessary as to ensure that the critical processes of the following kind are executed correctly:

- Operating with voting data
- Setting up the components
- Securing components or securely erasing data

4.4 - Data may only be transferred between the setup-component and the verifier by physical support which is then securely erased or the support securely stored.

4.5 - A number of voting cards must be compared with the verified file containing the parameters to be printed. This is to observe that the parameters are printed correctly.

Proposition B

1 - The notion of the printing service is replaced by “setup-components” and “printing components”.

2 - Functionality of the components

2.1 - Setup-components

- They generate the security-relevant parameters as defined by the cryptographic protocol, including the codes for individual verifiability.
- Unlike above, they can still generate secret values on their own. The generation of public values needs to be verifiable using a verifier as currently defined in the VEleS.

2.2 - Printing components (the machines that receive the file containing the parameters to be printed, as well as the printing machine)

- They verify the origin of the file containing the parameters to be printed (signatures of setup-component).
- They decrypt the file.
- If the file is not yet in a printable format, they create a printable file.
- They print the voting cards.

3 - Trust-assumptions (assuming one setup-component and one printing component):

3.1 - It is trusted that the setup component generates secret parameters correctly

3.2 - The setup component is trusted not to leak secret data.

3.3 - The printing component is trusted

- To verify signatures soundly
- To print according to the verified printing file
- Not to leak secret data

3.4 - Channels leaving the setup-component are untrusted.

4 - Operational requirements for “setup-components” and “printing component”:

4.1 - Before performing critical operations the following measures must be taken:

- Components have to be secured prior to operation
- Network connections must be removed physically
- The origin of the software to be installed has to be verified. The origin of the internet voting software has to be verified based on comparison with a signed published representation

4.2 - After performing critical operations the following measures must be taken:

- Components and any physical support holding critical data must be securely stored after use or the data securely erased
- Unless there is a significant reason, data is securely erased after use.

4.3 - Organizational measures need to enforce the consent and presence of a sufficient number of persons in the roles necessary as to ensure that the critical processes of the following kind are executed correctly.

- Operating with voting data
- Setting up the components

- Securing components or securely erasing data

4.4 - The functionality used in the software of the setup component for generating random values must enjoy special scrutiny.

4.5 - Measures must be put in place to ensure that sufficient entropy is used for generating random values.

4.6 - A number of voting cards must be compared with the verified file containing the parameters to be printed. This is to observe that the parameters are printed correctly.

Related Questions

The related questions are labelled Block-3.

Individual links to related questions

- Block 3 Discussion A - Assessing Proposition A
- Block 3 Discussion B - Assessing Proposition B

Questionnaire

The thesis is based on the questions 2.1.3 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.1.3	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 3A - Assessing Proposition A (Block 3 - Print Office)

Reference to originating discussion block

Block 3 - Print Office

Question

Assuming sufficient expertise at designing, implementing and scrutinizing a solution compliant with proposition A, how would you assess the added value? Which are important residual risks?

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to March 01, 2021 6 months ago

Christian Folini @christian.folini added Block-3 Cryptography labels 6 months ago

Christian Folini @christian.folini changed the description 6 months ago

Srdjan Capkun @Srdjan.Capkun · 6 months ago Developer

@christian.folini These two proposals are somewhat hard to compare. Is your question how to design a print office with enough robustness to deal with a compromise of a subset of its operations. In particular in the context of the compromise of the random code generation process.

We noticed that you contrast two designs, one that provides robustness by distributed generation of randomness, and the second one that relies on the verifiability of the randomness generation.

Or is your question more if robustness of print office to compromise is achievable and how.

David Basin @David.Basin · 6 months ago Developer

@christian.folini I think it has taken some time for the different groups to respond as the question is rather detailed but yet not all aspects of the design are fully clear. This makes the comparison difficult. Perhaps this kind of question would be aided by a zoom session?

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you guys and good proposal @David.Basin . FedCh will get in touch with you directly to clear things up. Based on this conversation the propositions / questions will be adjusted.

P.S. We'll launch the public bulletin board discussion block tomorrow, Thursday May 28.

Reto Koenig @Reto.Koenig · 6 months ago Developer

We honestly try to answer that question since quite some time, however, we cannot agree with the propositions at hand.

Our opinion on the subject is as follows:

Parameters

All security relevant parameters, an election is building upon, are already provided in the specification. For example, the value of

p,q,g etc are written down in the specification and the reason why it is those parameters is also provided. This way, the verification of such parameters is publicly available and truly open. Anyone can challenge these values and complain if something 'up any sleeves' is detected.

Secret Values

secret values that required in an election are protected by the 'one' group of trusted control components. Secret also implies not guessable, hence, high entropy. If at least one of the control components provided high entropy and keeps the distributed secret ...

- Long-Term ...for long term, privacy will be preserved (depending on the security setting). The same is true for the permutation of the mixing.
- Short-Term ...during the election, the correctness of the result can be guaranteed. (aka the per voter return codes, confirmation code, finalization code)

It is not possible to verify that a component really draws its random values from high entropy sources. One simply has to 'trust' it. Trust can be augmented by verifying the source-code that provides the random values, by operational measures such as making sure that /dev/random is not linked to /dev/zero, etc. but there is no final guarantee that the random values are unguessable for the attacker.

That said, we do not support any new component i.e. 'verifier' that tries to 'verify' the 'non-verifiable' requiring special trust assumptions. There is no option for a 'setup-Component' and its 'setup-verifier' on cryptographic / protocol level. Both of which state a single point of failure and that is bad.

We do not mind if some checks are made on operational level... as long as this does not jeopardise voter privacy or the correctness of the election, within the given trust model.

Printing

In cryptographers heaven, there is no central printing component. Every human voter will privately receive the secret shares from the 4 control components for individual verifiability. Then the voter will check their authenticity and will combine those shares in order to get the return codes, the confirmation code and the finalization code. Maybe by using a 'per voter' secure hardware device (definitely not the personal smartphone).

In the real world where usability is a major concern, the human voters are not capable of verifying the authenticity of the values and doing the required 'xor' operations themselves, and in Switzerland there is no such thing as a 'per voter' secure hardware device.

So, this secure hardware device must be made central -> Printing component. And all the printing component has to do is: Check the authenticity of the received secret shares of the 4 control components, combine the shares in order to get the various codes per voter and send these codes to the individual voter.

As the printing component cannot be split in a trusted group by nature, it states the single point of failure within the voting process. So, some 'bridge' over the 'air-gap' at the printer's side could (as always) breach the secrecy of the codes and pave the way for the perfect attack on correctness of the voting result, when the voting-clients gain knowledge about all the secret codes in advance.

This way, the printing environment (everything involved in the secrets) must be kept as simple and strict as possible. Only an input channel to the printing component, but no back channel... -> The printing component is an offline component. Secrecy of the codes has to be further protected by organisational means.

Please note, that after the election period, the printing component does not state any risk at all and thus does not have to be destroyed or locked away or anything like that. This is due to the fact that the codes used during voting loose their 'request of secrecy' as soon as the voter has finished the voting procedure. After that moment, the voting system can publish all codes of any voting card. Hence, every voter has the chance to check if all the codes on the printed voting card correspond to the codes provided by the control components. This states a supplement verification step for individual verifiability. Voters can send their voting card to the universal verifiers, so they can do this check as part of the universal verification. Be it one or the other way, the actions of the printing component can be verified as part of the complete verifiability.

If we are within the covert adversary model, the printer cannot do a 'denial of service' attack (by printing wrong values) without being caught.

That said, we do not support any additional online 'verifier' at the printer's side requiring special trust assumptions and not being part of the complete verification process. And the requirements (in terms of mathematical requirements) for the printing component seem bearable in order to not having to introduce new components.

Edited by Reto Koenig 6 months ago

Vanessa Teague @Vanessa.Teague · 6 months ago

Developer

@Reto.Koenig says, "In cryptographers heaven, there is no central printing component."

I would say "The central printing component is the thing most likely to send everyone to hell."

This is by far the hardest part of the whole protocol, and it's not clear that there is any solution good enough to form the basis of a trustworthy system. For parameter generation, there are probably fine solutions, but for the challenging issue of privately printing all those critically-important secrets, I don't see any good solution.

I think that's why there's an inconclusive set of answers to these propositions.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

In addition to the comments by @Srdjan.Capkun, @Reto.Koenig, and @Vanessa.Teague I would like to add that it is difficult to judge if the requirements given in Sections 3 and 4 are sufficient to render the components trustworthy. An idea might be to define a trusted base of computing (TBC), which must include the signature checking algorithm mentioned in Section 3.5, but what else?

As a side remark, requiring network connections to be removed physically may protect against a DDoS attack, but does not necessarily protect against malware that has entered the components when they were still connected. I was also surprised to read that only the origin of the software has to be verified and not the software itself.

David Basin @David.Basin · 6 months ago Developer

I had some problems understanding the scope of this. If one looks at the system used by Post/Scytl the setup is elaborate and includes generating a PKI infrastructure, generation of numerous RSA and ElGamal Key Pairs, secret sharing of keys, etc. So if we take the view that "printing" is now all setup operations, it wasn't clear to me where one draws the boundary. Also, the question assumes a particular architecture where setup-components generate security relevant parameters, verifies verify them, and printers print them. One could imagine other architectures. For example, it has been pointed out that private printing is problematic, so one could imagine splitting the printing office and having one office print and mail authentication codes and the other generating and distributing the codes for voting (e.g., choice codes, return codes, confirmation codes).

Regarding the architecture, as @Reto.Koenig pointed out, security relevant parameters can be part of the specification and published as part of it. This could be done in the specification of the protocol posted on the Bundeskanzlei's web page (no blockchain needed). Printing is certainly a concern. Whether the interaction between the setup component and the verifiers is sufficient (e.g., to preserve secrecy of private values) depends on protocol details. However, I did not understand the model in full, e.g., the significance of "3.3 With regard to verifying secret parameters generated based on values established by persons: It may be assumed that persons will not be able to memorize very long values, e.g. the hash of the other persons' passphrase." Where is this needed? And what humans can't remember they can take pictures of, especially if they are adversarial.

Edited by David Basin 6 months ago

Collapse replies

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

"This could be done in the specification of the protocol posted on the Bundeskanzlei's web page (no blockchain needed)."

[OS]By pre-defining parameters in the specification (and even publishing the specification), it becomes verifiable whether these pre-defined parameters have actually been used. But just being able to verify is not sufficient. If verifiability hinges of the correct usage of these parameters, the correct usage of these parameters would actually have to be verified (control-components and verifiers should only return a positive result if computations have been done using these parameters). Since pre-defined parameters could be introduced into the system components individually (into the untrusted and the trusted components) e.g. by hard-coding them, the setup-component and the setup-verifiers might not bring any added value in these cases. If this is so, the regulation should give room for such a solution. However, the secret values to be printed are a separate question. They cannot be predefined in a specification. The question is therefore, how to make the generation verifiable (to the degree that the values have been generated based on four random-sources) and how that generation should be verified. Since the printing takes place before the voting phase, it could make sense to verify before the voting phase, hence the setup-component and the setup-verifier. If it seems equally reasonable to perform the verification after the voting phase (e.g. by sending in the voting-cards and querying a "verifier" that received the individual contributions from the "control-components"), the regulation should allow that possibility too.

"If one looks at the system used by Post/Scytl the setup is elaborate and includes generating a PKI infrastructure, generation of numerous RSA and ElGamal Key Pairs, secret sharing of keys, etc." and "Whether the interaction between the setup component and the verifiers is sufficient (e.g., to preserve secrecy of private values) depends on protocol details."

[OS]I don't understand these two points. The protocol would have to be defined in such a way that if at least one of either setup component or setup verifier is honest, falsely generated values would be detected and secret data will not leak. The protocol details would need to be defined to satisfy that requirement, not vice-versa. This would be a major change to the Post/Scytl system, i.e. on protocol level (most values are not generated using 4 random sources, e.g. short return-code, confirmation-code, finalization-code ; let alone could this be verified today).

"However, I did not understand the model in full, e.g., the significance of "3.3 With regard to verifying secret parameters generated based on values established by persons: It may be assumed that persons will not be able to memorize very long values, e.g. the hash of the other persons' passphrase." Where is this needed? And what humans can't remember they can take pictures of, especially if they are adversarial."

[OS]The goal of having four random sources serves having sufficient entropy. This could be achieved by collecting the necessary entropy trough the contributions of four control-components (computers). In order to be open for any reasonable solution, we thought that it could also make sense to allow humans to contribute entropy (each of the four could choose a passphrase or roll dice). In a ceremony they could for example enter their passphrase into the setup component and again into the setup verifier.

[OS]Regarding forgetting values: If a ceremony would include looking at values on the screen e.g. for trouble-shooting, it does not seem appropriate to display a passphrase (although a passphrase might have sufficient entropy, it might be easy to memorize). To that end, the individual persons could privately compute the hash of their passphrase and confirm that the displayed hash is correct. We don't have any particular solution in mind, and there are high chances that this would not be needed.

[OS]Regarding taking a picture: The processes at the cantonal administration need to be trusted to some degree (like if the verifier says "no", the administration won't say "yes".) This has to be enforced by appropriate procedures. If private values are displayed, the procedure would need to entail ensuring that no one will take a picture unnoticed as well.

Edited by Oliver Spycher 6 months ago

David Basin @David.Basin · 6 months ago Developer

Thank you @oliver.spycher for the clarification!

Concerning: [OS]: The protocol would have to be defined in such a way that if at least one of either setup component or setup verifier is honest, falsely generated values would be detected and secret data will not leak. The protocol details would need to be defined to satisfy that requirement, not vice-versa. This would be a major change to the Post/Scytl system, i.e. on protocol level (most values are not generated using 4 random sources, e.g. short return-code, confirmation-code, finalization-code ; let alone could this be verified today).

I am wondering how you want to achieve that data would not be leaked in the case that the setup component is dishonest and the verification component is honest, but just performs checks. Do you have a concrete protocol in mind here?

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

Thanks, @David.Basin , for the question, it also made me notice that there is a mistake in 3.4: "The setup component or the setup-verifier is trusted not to leak secret data, UNLESS by encoding data into their output." It should say: "The setup component or the setup-verifier is trusted not to leak secret data BY encoding data into their output." Clearly both components can see private data which needs to be protected by organizational measures. However we want to get protection from secret data potentially being encoded into public data.

But your question makes me believe that you got right the meaning anyway.

Maybe it could work similar to this?

-The setup component saves only public data an USB key 1 and all data on USB key 2

-First, data from USB 1 is transferred to setup verifier (the setup verifier has not seen private data so far), then removed, afterwards USB 2. Based on USB 2, the setup verifier verifies the parameter generation and the consistency with USB 1 (data generation being verifiable, deterministic)

-If the setup verifier is honest but not the setup component, it will say that the data on USB 1 is not correct and it should not be used

-If the setup component is honest but not the setup verifier, the setup verifier could not save any private data to USB 1, because it did not hold any private data to save on USB 1.

In any case, this would need to be looked into in detail.

Edited by Oliver Spycher 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

@oliver.spycher : do you want me to fix 3.4 as outlined above?

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

Thank you all who have already posted statements to this block. This message is intended to give clarifications regarding the presentation of this block as well as some feedback for the further discussion.

Today the VEleS allows to consider the print office to be trustworthy. The VEleS is not precise about which operations additionally to printing could formally be performed within the print office. On the regulator's side we will need to provide clarity as to which operations seem permissible to be performed on a potential single point of failure and which ones do not. The regulation should be sufficiently concrete, so no unwanted solutions become permissible, and it should be sufficiently general, so no acceptable solutions are excluded.

One might say, the print-office should just print and that's it. However, that leaves the open question what the conditions for parameter-generation should be. This question certainly applies to the secret parameters to be printed, but one can also wonder whether the other parameters that have to be introduced into / chosen by the system should be regulated more rigorously.

It seems that most parameters can be generated and verified in a straight-forward way by implementing components that are already defined in generic terms in the VEleS. For instance: (1) today's control-components could generate their private share of the encryption key before the vote, and, in the context of verifying the correct decryption, today's verifier would also check that the public key was constructed according to the control-components' contributions. (2) other public values could be chosen in some verifiable manner, e.g. by copying them from the specification, and today's verifier would check whether this has been done correctly.

However, if the printing office only prints and does nothing else, it does not seem straight-forward how the secret values that need to be printed should be established and verified, given only the generic components identified in the VEleS today. The BFH-group propose an elegant solution where voting-cards are sent in for inspection against the values obtained from the control-components after the voting phase (correct me, guys). However, to me it is not clear whether verifying the voting cards (and other parameters) only after the voting phase (for the first time) would be early enough. Even if it would be considered early enough, I wonder whether Proposition A could be seen as being similarly effective with regard to verifying the secret values that need to be printed.

The main idea behind Proposition A with regard to generating the secret values to be printed: Different sources should be used in order to guarantee sufficient entropy. It seems inevitable to me to imagine a computer that aggregates the contributions from the different sources. This computer is called the "setup component". Since things can go bad if the secret values are not printed according to the contributions from the sources, the "setup verifier" is used to verify that the aggregation has been done correctly (e.g. by checking the signatures of the random sources and repeating all operations the setup component performed). This follows the logic of using separate independent components in case another component might be untrustworthy. The verified values are then used as the benchmark for checking the voting cards.

We are aware that different models for printing and parameter-generation are imaginable. As experts already pointed out, printing could be split up between different printing offices. We have brought proposition A to the table so we have something concrete to talk about and because we had the impression that it could generate significant added value. But it does not all have to be about Proposition A. Of course experts are always invited to share and discuss alternative ideas.

Bryan Ford @Bryan.Ford · 6 months ago

Developer

In general, I'm not really convinced that either Proposition A or Proposition B would provide a great deal of added security securely value. Either may still be worthwhile, if the cost is low - separating roles like this is certainly a good practice in general. But the fact that both Propositions still leave us with the printing office still being fully trusted "Not to leak secret data" is what really worries me most. It is especially for this reason that I would agree with Vanessa that "The central printing component is the thing most likely to send everyone to hell" - and neither Proposition helps much with that.

As I've pointed out elsewhere, any state-of-the-art modern printing facility is fundamentally dependent on electronic devices with complex hardware/software stacks and all the usual flaws and vulnerabilities. The fact that they're run offline doesn't convince me that that makes them safe, especially from a determined and resourceful state-level adversary. A printer's software or firmware could be hacked so it turns on its WiFi and when it's supposed to be offline and connects to the access point run by the attacker's black van outside. (Are the printers required to be not only offline but in a faraday cage?) An extorted employee might just plant a recording device where it can see and record the printer's output as the cards come out, and come collect it afterwards. A more sophisticated attacker could, I bet, come up with a sheet-feeder or toner-cartridge hack that makes the component seem normal but can record (and either transmit or just save for later pickup) the codes the printer prints. Alternatively, the adversary might not need to exfiltrate the codes at all if the printer can be hacked to make them predictable and known to the adversary, at least for a pre-selected list of voters whose devices the adversary has already pwned or knows it can.

Physical supply-chain attacks are relevant here too: the printing hardware, or computer(s) that will be decrypting the codes and sending them to the printer, might have been hacked/backdoored before shipping, or "diverted" during shipping. Of course these risks are ever-present in general, but the complete trust in a single printer with no hint of redundancy or diversity is the problem.

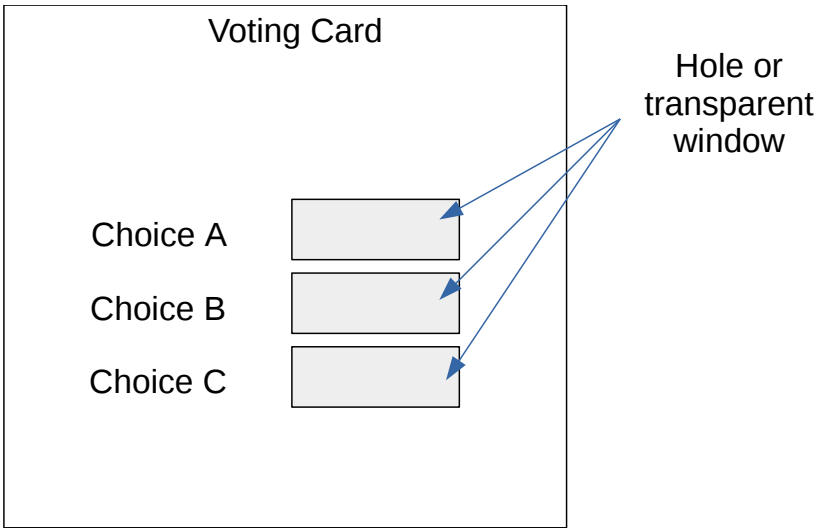
If a significant investment were to be made in strengthening the Print Office in the current E-voting system, I would try to find a way that can be done not just by separating roles, while leaving each role completely trusted in certain scary respects, but by doing proper parallel trust-splitting as is already done with the control components. Is there room for a "Proposition C"?

One approach that has already been suggested is to print multiple items (e.g., cards) and mail them all separately to the voters, so each voter needs to scan and use each one. That in principle could work, but there are major obvious usability costs and financial costs (e.g., doubling or quadrupling the already-considerable printing and mailing costs).

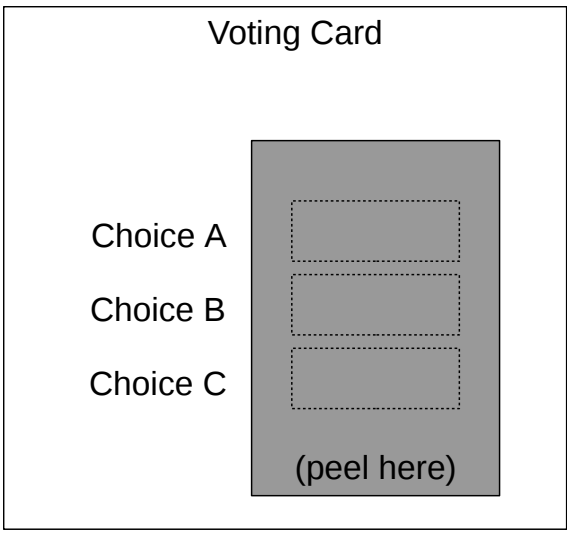
A perhaps lower-cost alternative for a "Proposition C" might be to achieve something like the "control component" splitting purely internally within the print office, so the print office manages to produce one card per voter without any single component in that office ever being able to see the (complete) secret voting code information that an attacker would need to change votes without detection.

As one no-doubt imperfect example of how this might work, I could envision a six-step printing process that looks something like this:

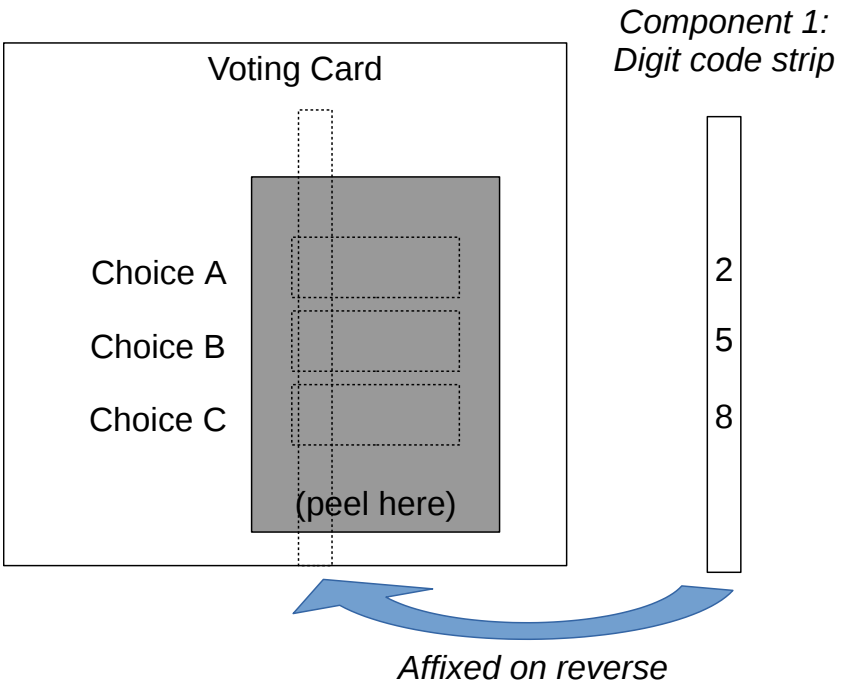
1. A less-trusted printing component at the print office is first responsible for printing the "backgrounds" of the voting cards with the candidate names and such. But instead of printing the secret per-choice codes themselves, this step only punches holes where the 4-digit per-choice codes will appear. So the result of the first step might look something like this:



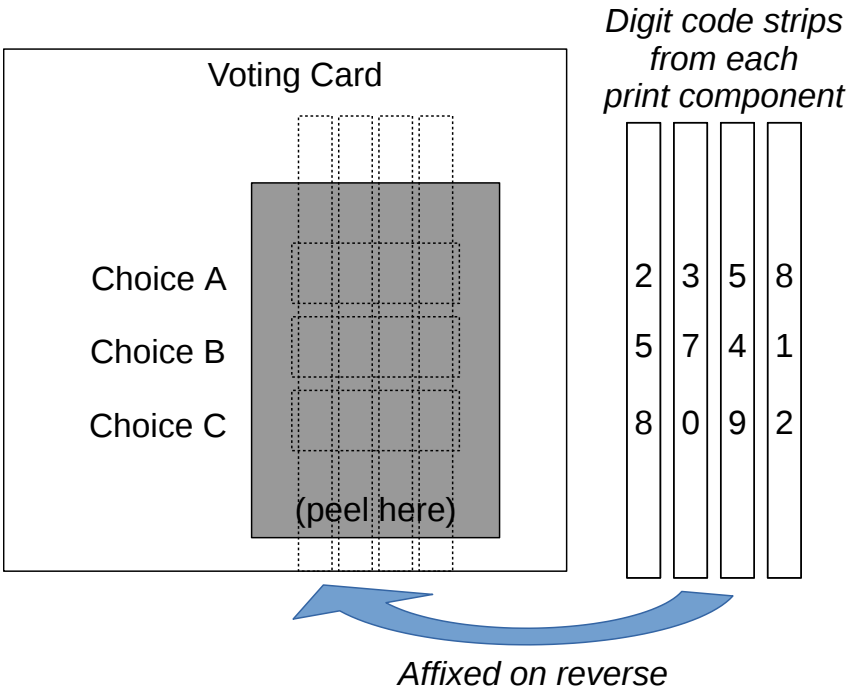
2. The print office affixes an opaque, security-patterned, tamper-evident peel-off sticker to the front of the voting card, which only the recipient user is supposed to peel off. Voters are instructed to report if they find the peel-off sticker missing or apparently tampered with when they get the card. So the front of the voting card looks like this with the peel-off stickers:



3. Now the card goes to the first high-security "printing component", which prints one vertical strip of digits - one digit for each candidate choices - and affixes that strip permanently to the back of the main voting card so that the digits appear through the voting code holes (but can't actually be seen until the peel-off sticker is removed):



4,5,6. The card then goes through the other three high-security "printing components", each of which ideally uses separate control and printing hardware, sourced from separate vendors via diverse supply chains, each was installed and is managed by a separate pair of administrators who have access only to one printing component, each in a separate secured space, etc. Each prints a separate vertical strip of one digit per choice and affixes it to the back, so the result is something like this:



This way, in principle, even if three of the print components are corrupt and leaking (or maliciously producing) their digits of their vertical strips, a single uncompromised component can ensure that voters who check their codes have at least a 9-in-10 chance of detecting a tampered vote because one digit doesn't match. A physically or electronically compromised print component, or a strategically placed spy-cam at its output, shouldn't be able to obtain the code digits produced by the earlier print components in the sequence because they're hidden under the peel-off sticker. Of course I have no doubt there are ways to read what's under a peel-off sticker (this isn't my area of expertise), but that's probably pretty hard to do with spy-cam style hardware let alone a software-only printer hack.

At a software and cryptographic level, this process might amount to the same thing as having four separate print offices generate four separate voting cards for each voter, each with only 1-digit rather than 4-digit codes. The only difference is physical-security techniques are being used to combine all four virtual voting cards into one before mailing.

Anyway, again this is just in the interest of exploring the design space. I have no idea whether it would work, how practical or impractical or expensive something like this would be, etc. And it still doesn't address the weaknesses resulting from the single postal *delivery* channel, which I can't see how to address without multiple mailings per voter.

Christian Folini @christian.folini mentioned in issue #29 (closed) 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

This is a very interesting proposal @Bryan.Ford. Thank you very much. I confirm it will be examined and discussed with the cantons and ideally also printing engineers.

The proposal is so interesting because it ignores today's printing facilities and looks at an ideal secure setup. It looks quite secure to me and I like it how you made sure that nobody ever gets to read the entire codes before the voter pulls away the peel-off sticker.

I reckon everybody agrees it is a quite advanced setup. If you have ever looked at a modern print office, then you know everything is about speed and scale. Multiple runs for the same sheet of paper is very difficult. There exists high-secure printing for items like credit cards (or bank notes), but in these situations there is a bank willing to pay the price for multi-pass printing for relatively few credit cards. With voting material, it's very large numbers being printed, so the costs need to be very low. That's why a single pass printing process is used nowadays.

In your proposal, the machines performing step 3, 4, 5 and 6 would need to identify the voting card, look up the right numbers and then print and glue the correct strip. This lookup is tricky and doing a "print strip 14,553 and glue it to the card in front of you" is not going to cut it since there is always something that can go wrong and it could ruin the whole series because it is not possible to check the correctness of the final voting material.

So the machines are simply not there to do this and we are in a troubling situation that way, because usage number for online voting are very low. You can hardly afford the development of a new printing machine for 2% of the voters.

However, adoption of internet voting will change that calculation, If 40% of voters use the electronic channel, the development of such a printer could become a worthwhile investment that results in a higher price per ballot, but the cantons might then be OK to actually pay it.

The two propositions A and B aim for an improvement for the current situation. That is: What can we do to ease the pain before we get to the high adoption rates that would warrant a completely different print setup like proposition C by @Bryan.Ford.

@Bryan.Ford seems to see some value in proposition A and @David.Basin also confirmed this (if I read him right).

Consequently, three questions remain:

- Do the other experts agree on the added value of A or B (or C for that matter)?
- Is there enough value to make it worthwhile improvement (or - assuming internet voting is not abolished over this very weakness - should the problem better be left alone until something better comes along)?
- Can you think of a simple mechanism that would bring an improvement for the current situation without costing too much? (In other words: Can you think of a cost-effective proposition that is better than A?)

Adrian Perrig @Adrian.Perrig · 6 months ago Developer

@Bryan.Ford's proposal is very interesting indeed. To perhaps simplify printing, one option may be to send a mask to the voter (printed and "hole-punched" by an independent entity), and in addition to send a regular paper with "superfluous digits" (printed by another entity). When the mask is overlaid on the sheet of paper, only the digits that need to be entered are visible.

Several options are possible here. The mask could be re-used across different voting cycles (although this has to be done very carefully to avoid determination of the mask holes), or the mask could be inserted separately into the same envelope by an independent printer or entity (to save on postal fees, but perhaps the post office would be happy to deliver two or multiple envelopes per voter).

Christian Folini @christian.folini · 5 months ago Maintainer

Let's try and round up discussion block 3, where we are trying to improve the security of the printing process.

I'm not yet ready for a summary, but here is what I have read:

There is a strong sentiment among the experts, that the printing office is a severe security problem and it is very difficult to fix the situation as long as we rely on today's printing process.

The two propositions A and B that are meant to bring some checks and verifications to the printing process were received with lukewarm approval so far.

Nobody sees a fundamental flaw with them, but the value they add is seen as small or modest at best. A likely conclusion is that an implementation makes sense if the costs can be kept small (unless a flaw is found in them).

Bryan Ford presented a fully blown proposal that involves multipass printing. It allows to create voting material with the full confirmation codes only visible to the voter ever. Let's call this proposal D. This proposal will need to be examined in detail, but after reading it a couple of times, it looks like a very interesting direction, even if it is unlikely to be doable with today's print offices (or the current budget to print the voting material).

Finally, Adrian Perrig proposed to print additional digits on the voting material and to have another entity print a mask with holes that makes the relevant return codes visible when laid over the paper with the superfluous digits. Let's call this proposal C. Proposal C is definitely easier to implement than proposal D. However, it probably does not bring the same level of security and comes with potential usability problems that might be bigger than with D. Still, it could be a useful step in the right direction.

All in all, the responses in block 3 have not been conclusive. 3B did not get a single response by itself after all. But the two new proposals are very interesting indeed.

The Federal Chancellery and the Cantons are planning to commission a separate examination of the two first proposals outside of the dialog in order to provide a near term perspective.

So this is what I have taken from this block. Please let me know, if I have misread something. I will now wait for a day or two and will then write up a summary.

Christian Folini @christian.folini · 5 months ago Maintainer

Good, no objection so far. Let me try a summary then. Proposition B / question 3B did not get a single comment and the discussion above goes beyond proposition A. My summary will thus try and round up the printing office completely.

There is a strong sentiment among the experts, that the printing office is a severe security problem and it is very difficult to fix the situation as long as we rely on today's printing process.

Two propositions A and B were presented to the experts. Both propositions are very close to the existing setup, but they improve parameter generation and introduce some verification without adding too much overhead.

The two propositions were received with lukewarm approval. Nobody saw a fundamental flaw with them, but only a few experts confirmed that these propositions add some value. A possible conclusion is that an implementation makes sense if the propositions can really be found to raise security and if costs can be kept small.

Adrian Perrig proposed to print superfluous digits on the voting material (proposition C). The return codes and the additional digits would form one long number. A second component would then print a mask with holes in it that corresponds with the voting material of the same voter. When putting the mask over the voting material, the relevant digits of the return codes become readable. The superfluous digits are hidden. This proposition could make it harder for an attacker to steal voting material via the printing process. However, it makes the printing more difficult, since multiple elements have to be produced and matched for the same voter. The delivery of the mask would need to be discussed and there might be a usability problem with putting the mask over the voting material correctly.

Bryan Ford presented a proposal that involves an even more advanced multi step printing process (proposition D). The return codes would be printed in multiple steps and glued onto the voting material in a way that makes sure only the voter will be able to read them. This proposition is much more complicated during the production, but it seems to reduce the usability problem with proposition C and also the transport is simpler than C.

All in all, the discussions around the printing office have not been conclusive during the dialog. Therefore, the Federal Chancellery and the Cantons are planning to commission a separate examination of the propositions A and B outside of the expert dialog in order to provide a near term perspective. In a second step, the propositions C and D will be examined as well. This discussion will also include representatives of printing offices in order to explore the mid- / long-term perspective.

The closing section is more or a comment than part of the summary. I'm adding it here, but it is likely that it will be moved into a different section or rewritten for the final report.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini mentioned in issue #17 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini removed Last-Call label 5 months ago

Christian Folini @christian.folini closed 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

May I attach a late addendum to my alternative design proposal above, based on subsequent discussion and further reflection? I'm not proposing that this issue be reopened, but just would like to add a few more potential points in the design space that may be worth exploring in the future in addition to the approaches A,B,C,D in the summary.

Voter-combined print components: One likely area of difficulty and expense in my earlier proposal (now "proposition D") that @christian.folini pointed out is the need for automated printing equipment to figure out which independently-printed components (e.g., code strips) go together for each voter, and align and attach them properly. This area of cost and logistical difficulty could potentially be mitigated by leaving the combination task to the voter. To summarize one way this might work:

- One independent print authority produces an opaque "base" voting card, in which only one digit of each code (e.g., the leftmost) is printed by that print authority and the other digit spaces are blank.
- Three other independent print authorities each produce transparent voting card *overlays*, much like overhead-projector transparencies, each of which is the same size and shape as the base voting card, but each has a different column of voting-code digits printed in the correct position for that column.
- Each print authority seals its voting card or overlay for a given voter in a separate, sealed, tamper-evident envelope, and mails it separately to the voter's address.
- Each voter opens the four separately-mailed envelopes, then manually places the three transparencies atop the base voting card, aligning them just like a deck of cards since they're all the same size and shape, so that all four digits of each code line up properly.

This approach would seem both logistically simpler for each of the printing authorities, and actually beneficial for security in a couple ways. The four print authorities no longer need to interact with each other "locally" at all so they can be truly independent, located in and mailing their materials from different companies based in different parts of the country. If multiple separate delivery mechanisms are available and not too expensive to use (e.g., postal service versus DHL versus local delivery from a nearby print office), then some diversity of physical *transport* means could be achieved as well. Even if all four print components are mailed to the voter via the same postal service, the fact that they are mailed in four separate events at different times and likely *from* different locations still gives them at least some useful diversity in delivery, which I think would go some way to mitigate many of the concerns Killer & Stiller studied and that we've been discussing.

Central mailing: If it is deemed too expensive to mail all four print components to each voter separately, then each independent print authority could first seal its component for a given voter in a tamper-evident "inner envelope", then one of the print authorities (or the mailing authority) collects all the inner envelopes for a given voter and mails them to the voter in a larger outer mailing envelope. This would presumably save on mailing costs, at the downside of re-introducing the need for someone to identify which four inner envelopes from the four print authorities belong together and should get mailed to the same voter.

Less trust-splitting: If either/any of the above alternatives are seen as potentially feasible in principle but still just too expensive because of the 4x cost increases, this could be reduced to more-or-less 2x cost simply by reducing the trust-splitting from four-way to two-way. For example, taking the independent-mailing proposal above, two independent print authorities (in different companies based in different parts of the country) each independently produce and mail to the voter one print component. One print authority produces and mails the "base voting card" containing two of the four digits of each voting code, and the other print authority produces and mails the one-and-only overlay containing the other two digits of each voting code. While clearly less of an increase in security and independence, I think this would still be far better than the current status quo of complete trust in both the printing and mailing authorities and no independent trust-splitting in this critical path.

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 3B - Assessing Proposition B (Block 3 - Print Office)

Reference to originating discussion block

Block 3 - Print Office

Question

Proposition B relies on one component generating values correctly. Do you see added value in adhering to the conditions in this proposition? Is there anything you would like to add?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to March 02, 2021 6 months ago

Christian Folini @christian.folini added Block-3 Cryptography labels 6 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

The summary for this question is done together with question 3A. Link

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Fixed links to questions for block-4

dune73 authored 6 months ago

e6f6fc9e

4-public-board.md 16.7 KB

Discussion Block 4 - Public Board

1. Introduction

Verifiability in Switzerland aims at detecting cases of large-scale fraud. Many computations cannot be verified by the voter but by the administrations or elected commissions. Their verification capabilities in return hinge on whether critical components they or their partners run are functioning correctly. Due to the trust-model defined in Art. 5 in conjunction with chapters 4.1 and 4.3 of the annex, it must be possible to justifiably assume that at least one control-component, at least one verifier and the printing office are functioning correctly. Refer to the introduction in Block 2 for details.

Vast scrutiny in connection with the number and diversity of critical components as well as separation of duty at operations serve the detection of manipulated votes. Theoretically, there is no maximum to the extent to which these practices could be implemented. In reality, the possibilities will be limited.

In this section we discuss an additional means of verifiability based on two propositions. The goal is to allow the detection of manipulations even if full sets of critical components of the internet voting system are not functioning correctly. The propositions are based on publishing vote-related information on a public system-independent platform (public bulletin board; PB). Voters are given read-access to the platform so they can additionally perform verification independently of the return-codes.

We would like to get your assessment of the added value in terms of security and trust-building. Also we would like to discuss the caveats that have been brought forward in the answers to the questionnaire, i.e. with respect to the long-term secrecy of the vote as well as the user-friendliness of the verification steps performed by the voters. Indeed, it seems that expert-opinions currently differ when bringing the upsides and the downsides into relation (see summary of 2.1.7 in the questionnaire).

2. Proposition A

2.1 In a nutshell

In this proposition the voters can use a PB for individual and universal verifiability. With this alternative they do not have to trust the voting system. The trust-assumptions are shifted.

Recall from the introduction to Block 2 that voters are instructed to report to the administration cases where the return-codes are not displayed correctly. At that stage, voters can then still vote by postal mail or at the polling-station. If the codes are correct, they can confirm the internet vote by entering their confirmation code. Proposition A naturally integrates into this process: Additionally to verifying the codes, the voters can verify the correctness of their vote and the acceptance of the confirmation code on the public bulletin board (PB). Thereby voters verify that their vote has been cast-as-intended and recorded-as-cast, i.e. without relying on the printing office or the control-components. After the voting phase, the results of mixing and decryption are also posted to the PB. Voters can verify that all votes (and in particular their vote) have been tallied-as-recorded. Voters can perform these verification steps an arbitrary number of times, possibly using different devices and different software. The parameters from the setup-procedure it takes to perform the verification as well as proofs of their correct generation are published on PB.

2.2 Functionality and trust-assumptions

The added possibility to detect systematic manipulations depends on whether a sufficient number of voters use at least one correctly functioning device at verifying the votes on the PB. The PB is designed to append signed messages received from the untrusted voting server and not to change or drop any content. The following trust-assumptions apply:

- The voting system (all control-components, verifiers and printing-office components) is untrusted regarding verifiability.
- The PB is untrusted.
- A voter-device used to verify the votes is trusted
- There is a trusted broadcast-channel that can be used as a boot-strap pointer to the PB and to serve as a foundation for detecting and claiming malfunctions of the PB (e.g. unpersonalized paper voting-material, official gazettes of the administrations, newspapers, maybe someday a block-chain).

2.3 Description

The PB and the software for the voter-device are meant to be implemented and operated as easily as possible, e.g. much easier than a control-component, and yet to bring added value with regard to detecting manipulations of votes.

1 - One or multiple server infrastructures are set up to constitute the PB. Each infrastructure returns either its full content, the content related to a voter (see 2c below) or the voting parameters at a given read-request along with a signed hash of the content. If the received contents differ between the infrastructures, the PB is considered not be functioning correctly and voters are instructed to inform the administration.

2 - The PB is initialized with

1. Encryption-scheme parameters, including the values that represent the voting options along with their text interpretation (e.g. question plus the answer or a candidate name).
2. Proofs of their correct generation.
3. Anonymous voter identifiers known to the individual voter (in current Swiss systems this could be a digest of the voting-card number which the voters enter to authenticate). Note that the detection of manipulated votes does not hinge on these identifiers or their pre-images being secret.
4. The PB's public signature key (a global one or one per infrastructure).

3 - A pointer to the PB (e.g. a URL) as well as a hash of the values on the PB are broadcasted through the trusted broadcast-channel. Based on this, the verification software is initialized and the public parameters are downloaded from the PB. (We assume for this example that the verification software runs on a mobile, the vote is cast on a laptop.)

4a - Before sending the vote to the server, the encryption of the vote and the random number used for encrypting the vote are displayed on the laptop, here we assume one QR-code, it could also be two QR-codes (one per value). The voters scan the QR-code with their mobile. Based on the random number and the parameters downloaded in step 3, the mobile displays the encrypted voting options as text. Voters can also use additional devices with different software to verify the encryption repeatedly. They cast the vote if they are satisfied with the verification result.

4b - Additional option: Prior to casting, voters have the option to create an arbitrary number of votes they do not intend to cast and verify the encryption of such a vote an arbitrary number of times. They can perform the verification with the same application on the same mobile or also with other devices possibly running different software. Once they are satisfied that the device they use for casting the vote encrypts votes correctly, they would create and cast the actual vote. It would be sufficient if the mobile only keeps the encryption of the vote, or in the case of two separate QR-codes, only to scan the one holding the encryption. The secret number would not need to be stored on the mobile. The contents of the QR-code might also be displayed as text for the voter to transmit to devices that cannot read QR-codes.

5 - Apart from generating and sending back the return-codes, the untrusted server signs the vote and posts it to the PB, assigned to the anonymous voter identifier. The PB accepts the entry if the signature is correct.

6 - Upon pressing a button, the application on the mobile phone directly downloads the contents from PB, signed by the PB. Based on the voting-card number (let's assume that in step 4 the voting card number was contained in the QR-code too), the mobile displays a text to confirm that the vote has reached the PB correctly.

7 - As in step 4, voters can repeat this step with other devices and/or other software. Once the voter is satisfied with the verification results, she enters her confirmation code.

8 - Apart from sending back the finalization code, the untrusted server signs a confirmation-message and posts it to PB, assigned to the anonymous identifier and the vote.

9 - Again upon pressing a button, the application on the mobile phone directly downloads the contents from PB and displays a text that the confirmation-message has been posted to the PB. This can again be verified multiple times using different equipment.

10 - After the voting period is over, the results from mixing, decrypting and possibly counting are posted to PB along with the proofs. The administration use the verifier to verify these proofs and additionally verify that the same votes are held by the control-components. In case of inconsistencies they start an investigation. Voters could also download the full contents and verify that the votes have been tallied-as-recorded.

11 - Server infrastructures that constitute the PB that change or drop votes prior to tallying will be detected by voters that read the full PB using a correctly functioning device. This however hinges on the assumption that there is at least one correctly functioning server infrastructure. Voters can use the signed content obtained in the previous steps to prove that their vote should be counted.

12 - If we want to assume that none of the server infrastructures should be trusted, the individual voters would need to verify that their vote has not been dropped or changed prior to tallying. To that end, using the information obtained in steps 4, 6 and 9, voters could locate their vote in step 10 and make sure that the encryption is correct. To address the case where all infrastructures show different content depending on who makes the request, the hash value of the full PB could be broadcasted through the trusted broadcast-channel. Again, voters can use the signed content obtained in the previous steps to prove that their vote should be counted.

3. Proposition B

In proposition A we assume that encrypted votes are posted to the PB. The encryptions will be decryptable in the far future. This could be addressed by posting perfectly hiding commitments to PB instead of the encryptions. This however would constitute a change to the existing Swiss systems which would require time. Further down we ask the question if it seems acceptable to yet post the encryptions, given anonymous identifiers. For the case where it might be concluded that this is not acceptable, we present proposition B as a possible intermediate mitigation.

In this proposition, the PB serves solely individual verifiability. However the public board cannot be used for decrypting the votes, i.e. not even in the long term.

To that end, not the votes are passed to the PB but a hash of the votes. The proofs of correct mixing and decryption are not posted to the PB. Thereby the benefit lies in the verification not relying on the control-components or the printing office to function correctly. It does hinge on the verifier functioning correctly. This might seem acceptable, given that it is running offline and within an observed process which is typically performed in the cantons' premises.

4. Related Questions

The related questions are labelled Block-4.

4.1 Individual links to related questions

- Block 4 Discussion A - Value of a Public Board
- Block 4 Discussion B - Public Board as a means to gain trust
- Block 4 Discussion C - Public Board and user-friendliness
- Block 4 Discussion D - Public Board communication
- Block 4 Discussion E - Public Board and voting secrecy
- Block 4 Discussion F - Public Board to determine voter turnout early
- Block 4 Discussion G - Public Board and support for voters
- Block 4 Discussion H - Public Board vs. Independent Control Component (see Block 2) and adapted parameter generation (see Block 3)

5. Questionnaire

The thesis is based on the question 2.1.7 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.1.7	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

6. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 4A - Value of a Public Board (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

Is a PB a valuable element of an internet voting system?

Drop or upload designs to attach

Linked issues 0

Christian Folini @christian.folini changed due date to April 01, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

David Basin @David.Basin · 6 months ago Developer

Let me check my understanding up front. The description says "The PB is designed to append signed messages received from the untrusted voting server and not to change or drop any content." The trust assumptions say: "the PB is untrusted". But this means that it can be malicious and need not fulfill its specification.

Trusting the bulletin board is too strong an assumption (why would we trust it?) but nothing good will come from an untrusted (malicious) bulletin board.

With Lucca Hirschi and Lara Schmid we looked at many voting protocols proposed using BBs, such as Belenios, Civitas, and Helios. If the bulletin board is untrusted, these protocols can all be broken, even if they offer individual and universal verifiability. In https://eprint.iacr.org/2020/109 we present attacks which in many cases amount to the bulletin board presenting different views to different participants independent of the BB's state (sometimes called equivocation). For example, the BB shows you your ballot when you check it for individual verifiability, but does not including it in the final set for universal verifiability. In general, the attacks can be quite subtle.

My concerns are:

1. I don't think "append only" (and not change/drop) will be sufficient as a property. Note that one must be very specific about the properties required by the bulletin board and the properties may be protocol dependent, i.e., one size BB may not fit all protocols. For example, if you wish to support vote-and-go, where the voter casts his/her ballot, and immediately afterwards performs Individual Verifiability checks (like described in block 2), then you would need (i) that the bulletin board's content grows monotonically (append only suffices for this) and (ii) also that there is some point where all parties agree on the BB's content. This second property is not entailed by the first one.

2. Given that an untrusted BB is too weak and a trusted one appears too strong, we need a replicated BB architecture that satisfies the required properties where only some fraction of the replicated BB components must be honest. There are different ways to do this (depending on the properties required and the distributed algorithm used) which yield different bounds on how many components must be honest. The paper can be consulted for technical details. Moreover, there is a very large body of literate on consensus protocols tolerating Byzantine failures.

The description in 2.3 (point 1) appears to allow multiple server infrastructures to constitute the PB, but apparently all must be honest as otherwise the voting process appears to abort. This seems too simplistic.

Collapse replies

Bogdan Warinschi @Bogdan.Warinschi · 6 months ago Developer

I want to raise a point related to @David.Basin 's remarks on the trust placed on BBs. Formal security models usually make implicit assumptions on the BB which are not always clear and almost never spelled out. These assumptions almost always include that the

BB displays the same view to all voters and that it does not drop or shuffle ballots around.

Besides the paper that @David.Basin pointed out, there is some additional work which tries to ensure that these assumptions are met. In particular there a sequence of papers starting with Culnane, Schneider which proposes a distributed BB implementation and some security properties and ending with the more recent work by Kiayias et al who propose a security model (in the UC framework) and distributed implementation for BBs.

In an upcoming paper https://eprint.iacr.org/2020/127 we show that even if the BB can tamper with the votes (i.e. drop or attempt to change/copy them) it is still possible to capture the level of privacy that schemes provides wrt to such attackers. Perhaps unsurprisingly, the level of security is related to the number of voters who perform the checks demanded by the voting system (the more voters check, the stronger the guarantees).

The point I want to make (in light of the above) is that it is important to demand that any formal model used in the analysis clarifies what are the assumptions placed in the BB so that these can then be checked against the particular BB implementation.

[I think @oliver.spycher hints that this may not be the best place for this discussion, but it's the place where the issue was first raised (perhaps we can move the discussion elsewhere if needed, but I didn't find a natural fit).]

Edited by Bogdan Warinschi 6 months ago

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

First of all, I'd like to highlight that this Block is about assessing the added value of using a PB at providing verifiability compared to not having it. It is not primarily about assessing the Proposition A itself.

In the answers, experts had differing opinions about PB. Proposition A is intended as a foundation to discuss the main issues of concern, in particular user-friendliness and long-term secrecy. Of course this only makes sense if the proposition itself delivers added value in terms of verifiability. However, it would be sufficient to agree that the proposition could at least in its essential points be implemented as illustrated in the proposition. It would also be sufficient to agree that this proposition actually doesn't work, but some other proposition would work that entails similar user-actions and similar efforts (financial, time) at implementing. If this does not work either, then I hope we could outline a different proposition and use that for the discussion. In any case, by no means is Proposition A intended to serve as a specification for an actual implementation.

At least for now, let's break things down based on the proposal.

Here are a few responses and questions related to the input of @David.Basin:

"The description in 2.3 (point 1) appears to allow multiple server infrastructures to constitute the PB, but apparently all must be honest as otherwise the voting process appears to abort. This seems too simplistic."

[OS] Let's relate this to the control-components. Instead of thinking about thresholds, I propose to accept that if one server-infrastructure misbehaves (let's call them PB-components and imagine they are operated at different sites), the voting process aborts. But manipulated votes should be detectable in the presence of misbehaving PB-components (in the sense of the security objective in Chapter 4.3 VEleS Annex).

[OS] Let's assume for a moment that we do not need the vote-and-go property. Would voters be able to detect a cheating PB due to the trusted broadcast channel (e.g. newspaper) in step 12 second part? If not, then why?

[OS] If yes, then let's now assume that many voters would not perform the verification according to step 12. Isn't there already added value if a fraction of voters perform that verification (assuming the cheating PB cannot predict which voters will perform the verification).

[OS] If vote-and-go should be supported, thereby possibly addressing voters who do not want to make a check after the voting phase is over but also do not want to rely on other voters to perform the check: Can you elaborate on your statement "(ii) also that there is some point where all parties agree on the BB's content" and explain what this would/could mean for implementing a solution (what does it mean, is it complicated to do, what can go wrong if it is not done)?

David Basin @David.Basin · 6 months ago Developer

Below are responses to the various points/questions that @oliver.spycher raises.

- [OS] "It would be sufficient to agree that the proposition could at least in its essential points be implemented as illustrated in the proposition."

Under the trust model (2.2 second bullet point), the BB can behave maliciously. You wouldn't want to implement it without replication. If you prefer to argue along point 12 with replication that is fine, but you would want more refined trust assumptions since if one of the replicated copies is dishonest, then denial-of-service is possible and you would have a loss of trust in the election process. It is possible to come up with an appropriate replicated architecture and trust assumptions (involving thresholds). So let's assume that is done and move on.

- [OS] Let's relate this to the control-components. Instead of thinking about thresholds, I propose to accept that if one server-infrastructure misbehaves (let's call them PB-components and imagine they are operated at different sites), the voting process aborts. But manipulated votes should be detectable in the presence of misbehaving PB-components (in the sense of the security objective in Chapter 4.3 VEleS Annex).

Yes. But do you really want to abort the election? In point 1 it is unresolved what happens here when the administrator is contacted. In practice, you would need a way to resolve disputes, since revoting at a voting station or by mail is presumably not an option (as it is, e.g., with Individual Verifiability Disputes).

- [OS] Let's assume for a moment that we do not need the vote-and-go property. Would voters be able to detect a cheating PB due to the trusted broadcast channel (e.g. newspaper) in step 12 second part? If not, then why?

If you really want to push the full PB to all users' devices and have them check IV on their phone at the end of the election, then you are right, they can detect this. Vote-and-Go would be more friendly though and pushing the full PB to the voters' phones is not particularly performant especially in cases where the PB contains many large Zero-Knowledge Proofs.

- [OS] If yes, then let's now assume that many voters would not perform the verification according to step 12. Isn't there already added value if a fraction of voters perform that verification (assuming the cheating PB cannot predict which voters will perform the verification).

I agree: there is added value.

- [OS] If vote-and-go should be supported, thereby possibly addressing voters who do not want to make a check after the voting phase is over but also do not want to rely on other voters to perform the check: Can you elaborate on your statement "(ii) also that there is some point where all parties agree on the BB's content" and explain what this would/could mean for implementing a solution (what does it mean, is it complicated to do, what can go wrong if it is not done)?

If there is not a point where all parties agree, then the adversary controlling the PB can show different content to different parties (equivocation). So there must be some final point, when the election closes, where, for universal verifiability, all parties can read the same content. Please see the paper with Hirschi/Schmid (https://eprint.iacr.org/2020/109.pdf) for examples of how popular protocols with Individual&Universal Verifiability can be attacked by a BB that does not mean the above requirement. In the paper cited, this property is called "Final Agreement". Combined with Monotonicity (related to what you call "Append Only") one has the properties one needs to make Verifiability "work" and it is not hard to develop protocols that achieve these two properties under certain threshold assumptions about some of the PB servers being honest.

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

This seems to me to be highly dependent on what protocol is used to implement the PB, and also what election protocol is designed to rely upon the PB. I completely agree with @David.Basin that if the PB doesn't achieve final agreement and monotonicity, it generally doesn't supply the assumptions necessary for most of the e-voting protocols that need a PB.

However, not all e-voting protocols make the same demands on their PBs. In some protocols (such as the vVote protocol that we designed for Victorian elections in Aus), the PB is simply a final transcript of accepted votes, with some mixing and decryption proofs. Voters have to check that their final vote is on it (so 'final agreement' is critically important), but they don't have to interact with it actively while voting (so 'append only' doesn't really arise).

Such assumptions can be based on distributed agreement protocols, but they don't have to be - for the simplified PB we used in vVote, we simply published a hash of the full contents in a local newspaper. Thus (though I never wrote out a formal reduction) the 'final agreement' property could be reduced to assuming that the local paper would be caught if it equivocated and printed different hashes for different people (assuming collision-resistance of the hash). Each voter's verification step consisted of checking that their vote was included on the website, and checking that the website's contents hashed to the value in the newspaper.

(And no this is not an advertisement for brilliant usability - I'm simply pointing out that, depending on the assumptions/needs of the voting protocol, threshold trust assumptions on the PB may not be necessary.)

Edited by Vanessa Teague 6 months ago

Oliver Spycher @oliver.spycher · 6 months ago Maintainer

Thanks, @Vanessa.Teague, I have a question regarding the PB used in Australia: Was there process to resolve conflicts where voters claim that their vote did not appear correctly on the PB? If yes, did they have to provide some form of evidence?

Reto Koenig @Reto.Koenig · 6 months ago Developer

The question from @oliver.spycher started some discussion on our side about dispute freeness of the e-voting system if the trust assumption (at least one component of the set of critical components is trustworthy) is dropped.

In the Swiss approach, if all four control components (and the printer) really collude, they can forge any valid ballot for any voter, as they know all secret values required.

This results in the situation, that the components cannot deny the ability to do any forgery including impersonation attacks. So,

every voter could pretend being the victim of such an attack. Hence, the protocol is not dispute free anymore.

For any e-voting system, where credentials are provided by trusted authorities (i.e. Swiss Approach with trusted control components) we claim therefore, that it is impossible to completely remove the mentioned trust assumption from the full set of the critical components, without loosing dispute freeness.

This way, we conclude, that the introduction of the proposed PB does not provide the possibility to remove the mentioned trust assumption. In contrast, the system owner must work hard to base the system in a way, that the mentioned trust assumption is very likely to hold... in order to induce strong believe.

It would be a different situation if a truly established PKI would be in place, where only the voter is in possession of its private key for signing.

Bryan Ford @Bryan.Ford · 6 months ago Developer

To start with, I think that in general there is great value in having a well-designed and properly-used public bulletin board in an E-voting system.

I agree with @David.Basin and @Bogdan.Warinschi that the precise goals, threat model, and properties of such a PB need to be clarified. I agree that both the "append-only" and "final agreement" (consensus) properties are almost certain to be needed in most reasonable uses of such a PB. My experience is that it is sometimes easy to imagine restricted/constrained uses of a bulletin board that don't seem to need these properties, but then you quickly run into trouble when you go a bit deeper, or change the assumptions or threat model just slightly, or make a tiny tweak to the protocol, etc. A PB abstraction with append-only and final agreement properties seems to be a fairly robust and broadly-useful abstraction; bulletin-board-like definitions without or with weakened versions of those properties tend to be much more fragile and harder to reason about.

Quite a bit of the above discussion seems to me to revolve around the question, "must a PB have trust-splitting across multiple components?" This of course depends on the threat model. In general, my observation is that you can get guarantees of transparency or tamper-evidence only if either (a) you make at least a threshold-trust assumption about the PB itself, i.e., that not too many PB components are compromised, *or* (b) you make an assumption that users (voters) have some form of side-band communication with each other not subject to manipulation or blocking even by a strong adversary that controls all the PB components. Two obvious examples of side-channel communication assumptions in the latter case are assuming/requiring publication of hashes in newspapers and such (and further assuming anyone actually reads and checks those hashes), or assuming some kind of independent peer-to-peer overlay network among users, such as those that blockchains like Bitcoin rely on (and further assuming that network-level attacks against that overlay don't succeed completely).

In practice, I see both of these threat models - a partly-trusted PB with no side-band communication assumptions and a fully-untrusted PB with side-band communication assumptions - as useful and complementary, but to maintain sanity and clarity it's important that we keep them separate. Further, I think it's useful and realistic to set the goal of designing a PB so as to satisfy both threat models at once, though analyzed separately. Some aspects of the design will be useful for one threat model but will seem useless when analyzed against the other, and that's fine.

In particular, in the "partly-trusted no-sideband" threat model, trust-splitting across multiple components is clearly essential, whereas in the "untrusted with sideband" threat model it looks like we could get away with a single, centralized PB server since it's untrusted anyway. In practice I think the benefits of providing at least threshold security in the absence of sideband communication (checking hashes in newspapers or assuming peer-to-peer connectivity) are easily strong enough to warrant this complexity cost, especially if the added complexity from the trust-splitting can be managed and minimized - and in my experience it can. For example, even though achieving the "final agreement" property across a threshold of components seems to imply Byzantine consensus, which tends to be hard and complex "in general", there are much simpler ways to achieve it in contexts that I think apply to E-voting systems like the one in question. For example, if you're using n-of-n threshold security where you want to tolerate n-1 corruption faults but do not need to tolerate any availability faults - the model the Swiss E-voting system's control components are already using anyway - Byzantine agreement is pretty easy. Even with a t-of-n threshold where t<n, Byzantine agreement can be easy when the agreement time-scales are slow and human-driven: e.g., if the only time an agreement happens is at the end of an election, when one of the human election administrators manually takes on the Paxos/BFT "leader" role and "pushes the button" to initiate agreement. There are important details to be considered carefully in here, of course, but they're manageable.

By having multiple PB components, in the "partly-trusted PB no side-band" threat model, the "vote-and-go" user gets a useful "immediate" guarantee of security (predicated on the partly-trusted PB assumption of course) regardless of whether he/she behaviorally ever reads newspapers, opens the voting app again after the election is over, has a device that participates in whatever peer-to-peer overlay network(s) might exist, or even has network communication at all or not after casting the vote. That's useful in practice.

But many people are understandably reluctant to assume completely that a threshold of PB components will never be compromised together or collude. So it's also useful in practice to be able to make guarantees that there is some way voters can "independently" detect misbehavior even if all PB components misbehave, which motivates the other, complementary, "fully-untrusted PB with side-band" threat model. In the worst-case where all PB components misbehave, it would be nice to ensure voters *can* also detect misbehavior by checking root hashes against a newspaper (even if in reality we know very few will do so), and/or if at least some voters' devices can check root hashes against those of others in an overlay network (or by checking the election outcome hash on Bitcoin or another public blockchain, which boils down to the same thing). The fact that these "independent" checks all definitely require side-band communication in some form or other doesn't make them useless.

And we clearly neither the "partly-trusted PB no side-band" nor the "untrusted PB with side-band" threat models are strictly

stronger/weaker than the other; they're incomparable and I think complementary. So my concrete suggestion would be to make this recognition explicit in the proposal(s), to state clearly that the desired PB design should be secure against *both* threat models independently, and to state clearly which features are relevant to one threat model but not the other (e.g., the existence of multiple components and the need for final agreement between them, versus the existence and dependence on side-band communication).

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

I have a fundamental problem with the notion of using a public board to verify results: I do not think that it is reasonable to rely on the voluntary participation of a poorly-defined subset of voters.

I think most voters would be confused by the public board and would not use it. An important question is also how the purpose of the public board would be explained to voters, and whether/how they would be encouraged to use it in the proper ways.

It seems to me that engagement with the public board would have to be channeled in a very structured way, for instance within the context of a Citizen Science project.

Just putting the the public board out there and hoping people will use it correctly does not seem a viable plan to me.

What am I missing?

(Sorry for being late in joining the discussion.)

Edited by Fabrizio Gilardi 6 months ago

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

I agree. It seems a bit far-fetched to me too. I cannot think of any analogous system as a precedent.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Still, isn't a secure PB a net positive? Under the assumption that all ballots cast (and only those) appear on the board a PB allows third parties to verify that the result announced does correspond to the ballots cast. This type of universal verifiability guarantees are not possible for the traditional paper-based voting and I can imagine that conveying this feature to the public may make evoting a more attractive proposition.

As @Fabrizio.Gilardi indicates, the verification procedure would most likely be be run by experts and not by the typical member of the public, but that should be sufficient: it sufficies that verification is carried out by a single member of the public.

Side note: the IACR (international association for cryptographic research) is running its elections online using Helios. There was one instance where (due to a communication error) the public bulletin board was not the same as the one tallied (I think by one ballot). Yet, there was no complaint which indicates that most likely none of the members has actually bothered to run the verification procedure. @Olivier.Pereira is probably more familiar with the details.

Edited by Bogdan Warinschi 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Bogdan, when we move the PB closer to the experts and further away from the public. Is it necessarily still a public board, or is it not something very similar to the verifier checking the control components from a conceptual point of view? (I'm not talking about cryptography, more about public scrutiny, transparence, governance, etc.)

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

That sounds like an accurate characterization to me. I think a public board is one which can be accessed, unmediated, by any member of the public (required expertise notwithstanding). If access requires some type of privilege then it is not a public board anymore, so it is closer to a control component. Universal verifiability is in this case something like delegated universal verifiability since it would rely on trust in one (or some) of the experts running the verification algorithm.

To understand the context of @christian.folini 's follow-up question: is it based on the assumptions that i) a public bulletin board does not go well with everlasting privacy and/or ii) a public bulletin board is essentially useless for individual verifiability and therefore it may make sense to pull the PB within the boundary of the system?

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your response. I was asking my followup question based on your response to Fabrizio Gilardi. The universal verifiability implemented so far is mandated officials (experts in a certain sense) running the verifier. This is very close to your response - outside of the fact, that they are not independent members of the public.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Ah, I see. But I have the impression that Fabrizio's remarks (like mine) did not concern the design where the PB is accessible to delegated experts/members of the public, but rather the one where the PB is accessible/used by members of the public directly. I agree that the PB is virtually useless for individual verifiability (for all the reasons identified) but I think it is valuable for universal verifiability.

Christian Folini @christian.folini · 5 months ago Maintainer

I'm with you 100%.

Before, I was not sure that my impression that there was a link on the concept level was correct and if I got this right.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

To make sure that there's no confusion (my statement above may be too strong) let me qualify that by "useless for individual verifiability" I mean that I would certainly not expect for a general member of the public to understand precisely *how* the protocol uses the PB to provide individual verifiability.

However: a) one can imagine that the more engaged members of the general public can be guided toward building an accurate mental model which clarifies how the PB is used in a way that preserves vote secrecy while helping with verifiability. (e.g. @Florian.Egloff's proposal below seems to be a concrete proposal on how to move in this direction) b) on purely technical terms, a PB is clearly useful since at the very least, it enlarges the design space of protocols with a powerful tool we can understand and know how to leverage.

Edited by Bogdan Warinschi 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

Yes, that is what I am suggesting. My intuition is that if we are going to introduce internet voting, it is important to have a roadmap for how you introduce such technology to voters. It bears repeating that you would want to ensure that all those elements designed into the technology that are supposed to make it trustworthy need to be introduced to voters, understood, and then used. Whilst we all acknowledge that a fraction of the total population will be interested, I would suggest that a broad outreach strategy gives you the maximum chances for the technology to be understood and trusted.

Having physical representations of how the technology actually works is one way to make it "touchable". Ideally, you would want to get the intuition of why & how it works across to a wide cross-section of the population. That is why I suggested "Gemeindeversammlungen" as a good place to do this.

Florian Egloff @Florian.Egloff mentioned in issue #25 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

The discussion in 4F (early publication of voter turnout data) brought up several interesting points that we will cover in the summary of this question here (4A) as this is also meant as an overarching discussion of public boards.

See here for the important items in 4F.

Florian Egloff @Florian.Egloff · 5 months ago Developer

@David.Basin raises dispute resolution as the key (here #25 (comment 922)).

I would expand on @Fabrizio.Gilardi 's comments: An attacker attacking the legitimacy of elections only needs to attack the appearance of trustworthiness of the process or outcome, not the actual trustworthiness. If the elections can be made to appear to be “rigged” then the attacker has a chance to disrupt the fundamental value in democracy, namely, that the losers accept the outcome. Internet voting has the potential to introduce such a means of sowing doubt. This has nothing to do with the objective “security” of the system, but rather, with the social familiarity & trustworthiness of the system by the population: if the population does not have trust in the dispute resolution before the election takes place, it will be hard to convince a population of their "trustworthiness" in the actual event.

The relevant question thus is not “do you trust internet voting enough to use it?” but rather, “given a scenario of widespread allegations of abuse and cheating, do you trust the process designed to prove to you the trustworthiness of the process and the result?” Thereby, I assess, it will be very hard to convince an (adversary stoked) distrustful population that one should trust the authorities’ claims that the technical systems worked properly, that the programs claimed to be running were running, and that the outcome is correct. This is not a remote possibility, but rather the baseline threat any change of democratic voting has to deal with.

Thus, I suggest, we should discuss the issue of dispute resolution, and the process of how voters form trust in it, here.

[Moved here from (#25 (comment 952)), since @christian.folini wants us to discuss remediation here (see #25 (comment 961))]

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree in general that social trust of the voting system (of any kind) is extremely important in general, and having a strong dispute resolution process is a critical element supporting that trust.

This issue has both technical (including cryptographic) and non-technical (e.g., social) implications. On the technical side, as I mentioned before, the security properties that a voting system (and especially a PB) should have must include a "no false accusation" property of some kind. That is, it should be provably infeasible cryptographically for an attacker to fabricate convincing digital "evidence" that the PB misbehaved, or that a user's vote was miscounted or left uncounted, when the control components are in fact operating correctly. This property is closely-related to (really just an extension of) the unforgettability proofs that digital signature schemes and general zero-knowledge proof systems must satisfy in order to be considered secure. If this technical "unforgeability of false evidence" property is defined and implemented correctly, then it *should* ensure that if someone (either maliciously or just naively) claims that the voting system is misbehaving, and a competent and honest security expert closely examines that claim, the expert will find that claim to be invalid and help debunk it.

The other big part of the dispute issue of course is social: why would we expect an average, technically-unsophisticated voter to trust the opinion of some random (purported) security expert against (say) an acquaintance or political figure who is spreading conspiracy theories that the voting system is misbehaving or otherwise untrustworthy? Let's not delude ourselves into assuming that most ordinary voters will (or even should) place blind trust in a group of distant supposed experts they've never heard of - especially experts chosen by the government, such as those of us participating in this dialog.

But this is where the *public* verifiability of a voting system with a correctly-functioning PB becomes crucially important and extremely powerful: it is verifiable not just by a select group of government-chosen experts, but by *anyone* who a particular voter might trust to advise them. Technology is so pervasive and unavoidable now that even the most non-tech-savvy (or expressly "technology-avoiding") users tend to have a friend or relative they know, or a neighbor's geeky kid, who they go to to ask technology questions like "can you fix my Windows laptop?" or "should I trust the E-voting system?" And that geeky neighbor kid, though he/she probably isn't a cryptographer or e-voting expert her/himself, is perfectly capable of answering, "hmmm, well according to the latest on Slashdot and Reddit this conspiracy theory has been thoroughly debunked by the experts, although they do say you need to keep your laptop software updated - oh it's that old, can I help you update it?" That's social trust: giving everyone their own *choice* of expert(s) to listen to. A voting or E-voting system's approach to social trust should ensure that all reasonable and honest people with reasonable social networks will be able to find good information in their own "social neighborhoods" that they can trust, and that that information will largely point the right way because the [E-]voting system is sufficiently transparent and open to all for inspection and verification.

Of course there will always be some die-hard conspiracy theorists who will only believe what other conspiracy theorists say no matter what, or who only trust self-proclaimed "experts" who are in fact just self-interested charlatans, operators paid or otherwise incentivized by an adversarial actor to sow public doubt and uncertainty, etc. And in highly-polarized populations like we see in certain large countries outside of Switzerland, identity politics may make it effectively infeasible ever to get most members of one party to trust anything that's perceived to come from the "other" party. But fortunately that extreme state of polarization does not appear to be prevalent in Switzerland. For this reason, I see reason for substantial optimism that at least here, social trust effects will come down generally in favor of (rather than against) a well-designed and truly-transparent, publicly-verifiable E-voting system design.

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

> That's social trust: giving everyone their own *choice* of expert(s) to listen to. A voting or E-voting system's approach to social trust should ensure that all reasonable and honest people with reasonable social networks will be able to find good information in their own "social neighborhoods" that they can trust, and that that information will largely point the right way because the [E-]voting system is sufficiently transparent and open to all for inspection and verification.

I like the idea of finding good information in your own "social neighbourhood". I would thus encourage some explicit reflection on how that comes about, i.e. what process one uses to ensure that such expertise is easily available (I believe having a good website is insufficient).

The idea I came up with – and I am sure there are many other ways of doing this – is this:

One idea would be to work with communal councils (Gemeinderäte). Gemeindeversammlungen are great places to reach the interested voting public. One could start by building touchable systems: i.e. replicate the system with physical objects for a workshop-like experience. Give people the option to “touch” the system logic. Explain how trust in the system is established & give people the option to “practice” internet voting right there. (quoted from #22 (closed): #22 (comment 988))

Florian Egloff @Florian.Egloff · 5 months ago Developer

@Bryan.Ford said:

> But fortunately that extreme state of polarization does not appear to be prevalent in Switzerland. For this reason, I see reason for substantial optimism that at least here, social trust effects will come down generally in favor of (rather than against) a well-designed and truly-transparent, publicly-verifiable E-voting system design.

I agree with you that Switzerland, with its less polarized politics, is better placed for such trust to arise. I would add that this does not replace the political work that will be necessary across the party spectrum to build such trust. In my opinion, and I am sure @Fabrizio.Gilardi might be able to be much more specific about this, this would require for the issue of internet voting to move out of the (technocratic) expert-driven space into one where political actors drive the (non-)adoption. I say this as I see the risk that @Bryan.Ford raised in #19 (closed), that if there is no such broad political support, the issue can be much easier be politicized adversarially.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

Many thanks @Bryan.Ford for this very helpful summary. I think I agree with everything. What still bothers me, though, is that we're making informed guesses regarding voters' engagement with the PB. I'd feel more comfortable if the PB had been tested in a realistic setting (i.e. an e-voting trial in Switzerland) and we had some evidence to rely upon.

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

Fully agreed that systematic usability testing and refinement is extremely important, and should be worked into the design, development, testing, and (gradual) deployment plan for a PB from the start.

Bryan Ford @Bryan.Ford mentioned in issue #37 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #8 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We have adopted all summaries for the questions in block 4 outside this overarching one in 4A. I'm repeating all other summaries here, since they add context to the summary for 4A, that I am going to propose.

I'd rather avoid to re-open the summaries of 4B through 4H for discussion. They are here for reference only.

4A : Value of a Public Board

Is a PB a valuable element of an internet voting system?

A Public Bulletin Board can be a valuable element of an internet voting system when several aspects receive the necessary attention during the design, implementation and introduction phases.

The list of these aspects consists chiefly of the following items:

- The Public Bulletin Board has to be designed carefully to fit into the overall design of the internet voting system.
- The trust assumptions should not be too weak and also not too strong. Using the appropriate threat models is an essential step.
- A vote-and-go capability would be helpful. Final agreement and monotonicity seem equally important.
- Any doubts with regards to the secrecy of the vote could have a very negative effect on the internet voting system as a whole.
- Dispute resolution is very important since an incorrect claim of voting fraud with reference to the Public Bulletin Board has to be refuted convincingly to experts and the general public alike.
- Communicating the role and the proper use of the Public Bulletin Board towards the voters is a very big undertaking on its own. This should involve ways to make the Public Bulletin Board and internet voting as a whole tangible for the voters so they can actually understand it.
- Extensive usability studies are essential for the success of the Public Bulletin Board.

- With regard to trials with Public Bulletin Boards, cantonal particularities should be taken into consideration.

Under the line, the Public Bulletin Board has the potential to improve security and trust. The security improvements strongly depend on how the board is implemented, used and the information displayed. Other than that, there are additional security risks that need to be addressed.

4B : Public Board as a means to gain trust

Is a PB suitable to establish trust?

A Public Board can establish trust. If done right, the evidence and the transparency a Public Board provides will have a positive effect on trust of the voters into the system, namely in a long term perspective. A Public Board signals the authorities' commitment to transparency, which in turn has the potential to translate into increased public confidence into the electoral process.

It is very important to note, however, that a misdesigned, a poorly implemented, a badly documented or an otherwise not optimally supported Public Board will have a negative effect on the trust: Unless the Public Board and everything around it, is of a very high quality, it could very likely have a negative effect on trust. So the Public Board can only unleash its value if the simple metaphor is supported by good design, proper implementation and adequate operation that also convinces the experts.

It is likely that relatively few users would use the Public Board themselves, but instead rely on the checks executed by independent experts / political parties. But this is not necessarily a problem and one could think of establishing such a process officially, while still giving users access to the Public Board directly or via official and independent tools.

The Public Board is closely tied to the tools that make use of it, since very few voters will use it directly. The interface between the Public Board and these tools is therefore very important. If there is a failure with this interface, the Public Board will be broken from a user's perspective. That's why changes to the interface have to be done in a very careful way and the tools and their providers have to be supported adequately.

4C : Public Board and user-friendliness

Does it seem likely that already or in the future a sufficient proportion of voters will use a correctly functioning device to perform the verification? How do you assess the potential user-friendliness?

It is very clear that user-friendliness is an important factor that decides on the degree of use by voters and also influences the proportion of voters that verify attentively enough to notice errors with the verification and also care to report them. However, requirements for user-friendliness in the context of internet voting must be specified before any thorough examination can be performed.

User interface studies are needed to assess various design and usage scenarios. It is also an area where testing of a design in a structured context like a citizen science project possibly in conjunction with e-voting trials could help studying the use of the public board and contribute back to its design.

Even if a mature and user-friendly design of the public board could be reached, there would still be the chance that an attacker could corrupt any particular voter's device(s) used for voting and / or verification. Likewise with social engineering attacks to steal secret information from the voter. It takes a sufficient proportion of voters that are non affected in order to detect large-scale fraud.

4D : Public Board communication

How should this additional method for verifiability be communicated towards the voters and brought into relation with the current method, knowing that up to now voters were told that only the electoral board is able to decrypt the votes?

Effective communication with the voters is important. This is particularly true with regards to good practice when performing the verification (e.g. different device, separate network, bluetooth turned off) but also with regards to possible security risks that emerge from offering a new means of verifiability.

It would be useful if a scheme could be developed that does not require to display information (e.g. randomness) that is critical for the secrecy of the vote.

The difficulty to communicate the proper use of a public board effectively, makes it a hard to solve problem.

4E : Public Board and voting secrecy

How likely does it seem that in the far future decrypted votes will be correlated with the voters' identities? Would we need to consider this a problem from societal perspective (as by now voting secrecy is guaranteed by Swiss law)?

Voting secrecy is a basic right, guaranteed by law. If a Public Board raises doubts about this guarantee, then this undermines not only trust in the Public Board but also in the internet voting channel and voting as a whole. The use of a Public Board raises some concerns regarding vote privacy. Depending on the cryptographic techniques used, advances in cryptanalysis, or due to bugs, the information on the board may reveal how a voter voted.

We can only speculate on the potential motivations for future attackers to attempt decryption or de-anonymization of past votes, and what harms such attacks could lead to, such as voter embarrassment or coercion. Nevertheless, it is important to understand the risks and trade-offs associated to using a Public Board, and different designs thereof.

Some Public Board designs publish encryption of votes. Here, secrecy of votes relies on the same set of assumptions which guarantee secret communication over the internet. Yet, these assumptions may be invalidated, and votes may be decrypted if quantum computers become a reality. If the board contains voter identifying information (e.g. required for auditing purposes) then the link between voters and their vote may be revealed. These designs may potentially be strengthened by using encryption schemes which are secure even against quantum computers -- such schemes are under development by the cryptographic community.

Other designs may hide the link between voters and their encrypted ballots so even if ballots get decrypted individual choices stay secret.

Finally, other designs aim to achieve "everlasting privacy". Instead of publishing encryptions of votes, such schemes publish only a so-called "perfectly hiding commitment", which registers the vote on the bulletin board, analogous to a hash of the encrypted vote, but which provably contains no information about the vote's content that could ever be decrypted or revealed no matter how powerful the attacker is. Of course, the overall guarantees for vote privacy still rely on the security of the rest of the building blocks.

4F : Public Board to determine voter turnout early

According to both propositions, the full contents of the PB would be readable by the public at any given time. Thus, the preliminary turnout would be known before the voting period ends. This could be avoided by limiting access to the parameters and to the data related to the individual voters before the voting period ends. However, we wonder whether knowing the preliminary turnout should be considered an interesting feature rather than a problem. What do the social scientists think?

It is very important, that any form of Public Board does not leak preliminary results of a vote or election. Data, that allows to deduce voter turnout is less of a concern, though. Also because this exists for voting by mail in some Swiss voting circles already.

4G : Public Board and support for voters

Following the logic of querying a PB with independent software of own choice, would it be correct for the cantons or the provider not to offer support in case of problems during the verification process?

It is important to distinguish different support situations:

1. Support with using a verification software that is officially endorsed by the canton.
2. Support with using a 3rd party verification software.
3. Support for a voter when the software fails to verify the vote and a dispute arises.

(1) A canton needs to provide support for voters using officially endorsed software.

(2) In case of 3rd party software the canton should maintain a list and in case of problems the canton should be able to direct the voters to the provider supporting the software.

(3) If the role of the Public Board is to verify the correct operation of the vote by the system provider and the canton as well. Then the canton is not the best contact in this situation and there has to be an independent support provider. However it is difficult to design a solution that addresses this case adequately: If the bad reports are escalated before they are thoroughly confirmed, then the trust could be undermined without good cause. And if the support provider discourages the voter from pursuing a report, then a malfunction or fraud could go undetected. It is therefore important to design and implement proper dispute resolution for this and other situations where a dispute may arise.

4H : Public Board vs. Independent Control Component (see Block 2) and adapted parameter generation

Under the line, the Public Bulletin Board has the potential to improve security and trust. The security improvements strongly depend on how the board is implemented, used and the information displayed. Other than that, there are additional security risks that need to be addressed. *Recalling Block 2, would you recommend Bob to introduce a PB or rather to procure new control-component software or adapt the protocol to achieve parameter generation that is more trustworthy, assuming he could only pick one?*

Few experts were willing to address the hypothetical choice between more software diversity, more trustworthy parameter generation and the introduction of a Public Bulletin Board. One could conclude that it is a very hard choice or that it is a very hypothetical one.

Software or more generally implementation diversity brings a quantifiable benefit in security with linear costs. As discussed in 4A und 4B, a Public Bulletin Board can be a valuable element of an internet voting system which can establish trust.

I'm aware that the PB question is very controversial. So let's see if we can nail it down until the platform closes.

[EDIT] Reworded several items based on input by @Bryan.Ford below.

[EDIT] Several typos spotted by @Carsten.Schuermann.

[EDIT] Cutting down on 4H to remove some redundancy on request of @barbara.erni.

[EDIT] A more precise version of the text from 4H was appended to the end of 4A on request of @oliver.spycher.

[EDIT] Split the bullet point on the trials and cantons on request of @oliver.spycher and reworded it to make it more general.

Edited by Christian Folini 4 months ago

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

I would change "the right threat model as a guiding line" to "appropriate threat models" - for one thing, because there is not necessarily a single "right threat model". In particular, we identified at least two different complementary threat models, neither of which formally subsumes or dominates the other, and both of which the system should probably satisfy (see this comment of mine and the discussion leading to it).

Nitpicky writing suggestion: "designed in a very careful way so it fits into" -> "designed carefully to fit into"

I would break this into two separate bullet points at the comma: "A vote-and-go capability would be helpful, final agreement and monotonicity seem equally important." Or if you prefer these to be in one bullet point about required properties, then just break them into two sentences in the same bullet.

And perhaps change "important" in the second resulting sentence to "essential".

The item about "Any doubts with regards to the secrecy of the vote..." doesn't make it immediately clear how this statement relates to a PB. How about, "The PB must be designed carefully to avoid increasing risks to the secrecy of the vote, which could have a very negative effect on the internet voting system as a whole."

Change "since a claim of voting fraud" -> "since an incorrect claim of voting fraud". (A *correct* claim of voting fraud should *not* be refuted!)

And maybe "in a credible way" -> "convincingly to experts and the general public alike"

But looking good in general, thanks!

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Bryan.Ford . Let's tackle this one by one:

1 - threat models: "the right threat model as a guiding line" -> "appropriate threat models"

Makes sense. Adopting your updated wording: ""

2 - wording: "designed in a very careful way so it fits into" -> "designed carefully to fit into"

Thanks. Much more elegant. Adopted.

3 - Break bullet point into two ("A vote-and-go capability ...")

I'd rather keep it together, also since the 2nd half refers to the first one. But I agree, it's better in two sentences.

If there are more people preferring to split this into two bullet points, then I'm OK with that of course.

4 - "equally important" -> "equally essential"

I think this is giving it even more weight and given the *equally* refers to *helpful* this would be odd. Maybe the whole bullet point should be reworded.

Open to opinions or suggestions.

5 - secrecy of the vote

I see your concerns, but I'm afraid replacing "any doubts" with "increasing risks" is reducing the weight, while "any risks" would be too high (after I have read through state statements again).

Christian Folini @christian.folini added Last-Call label 5 months ago

Other than this, I prefer your wording, but I think the "any doubts" is well balanced and the term is also not technical (on purpose!).

Christian Folini @christian.folini mentioned in issue #60 (closed) 5 months ago

Other opinions?

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

1. I think this modification is fine, except that now the summaries no longer relate PB to potential security benefits. I propose to make the modification, but in exchange to add a line to the summary in 4A which I believe finds at least implicit support in many statements: “PB has the potential to improve security and trust. [Then similar as in Olivier’s statement #20 (comment 1321)] The security improvements strongly depend on how the board is implemented, used, and on what is displayed on it, and there are additional security risks that need to be addressed.”
2. “Extensive usability studies are essential for the success of the Public Bulletin Board. In this phase, it should become clear if a Public Bulletin Board is an optional or rather a mandatory addition for the cantons.” -> Whether or not measures become mandatory depends on various considerations, in particular the added value to security. I believe that leaving PB optional to the cantons was proposed under the premise that PB is not strongly related to security (#25 (comment 880)). So I propose to replace the second sentence with a new line: “With regard to trials with PB, cantonal particularities should be taken into consideration.”

Christian Folini @christian.folini · 5 months ago Maintainer

It is getting late and we still have several proposals to update the summary on the table.

Proposal 1 (by @barbara.erni)

Idea: Updating 4H in order to remove redundancy and ambiguities between the different responses.

Old 4H:

> Software or more generally implementation diversity brings a quantifiable benefit in security with linear costs. The introduction of a Public Bulletin Board is a significant step that comes with big security improvements. On top, a Public Bulletin Board also brings greater transparency and thus an element that allows to improve public confidence in the internet voting system as a whole.

New 4H:

> Software or more generally implementation diversity brings a quantifiable benefit in security with linear costs. As discussed in 4A und 4B, a Public Bulletin Board can be a valuable element of an internet voting system which can establish trust.

This update is supported by @Olivier.Pereira, @David.Basin, @Bogdan.Warinschi as well as @oliver.spycher if I interpret him correctly.

Proposal 2 (by @oliver.spycher)

Idea: If the overly general security benefit is removed from 4H, reword it to be more precise and add it to 4A.

The point is that the question for 4A is very narrow focusing on an abstract value. The moment we made the 4A question an overarching question for block 4 (in order to retain the focus in the other questions of the block), it becomes more important to read the *value* in terms of security. This should thus be explicit. I slightly reworded Oliver's text for better readability.

Addition to the end of 4A:

> Under the line, the Public Bulletin Board has the potential to improve security and trust. The security improvements strongly depend on how the board is implemented, used and the information displayed. Other than that, there are additional security risks that need to be addressed.

This update is supported by @Oscar.Nierstrasz.

Proposal 3 (by @oliver.spycher)

Idea: The idea to make a PB an optional feature for the cantons only makes sense when security does not matter. As soon as there is a security benefit, it can no longer be optional. It is therefore better to make the wording more general so that particularities of the cantons are no longer focused on that aspect alone.

Old 4A:

> - Extensive usability studies are essential for the success of the Public Bulletin Board. In this phase, it should become clear if a Public Bulletin Board is an optional or rather a mandatory addition for the cantons.

New 4A:

> - Extensive usability studies are essential for the success of the Public Bulletin Board.
> - With regard to trials with Public Bulletin Boards, cantonal particularities should be taken into consideration.

This update is supported by @Oscar.Nierstrasz.

I think all three proposals are reasonable and all three seem to have some traction. So I am inclined to adopt them. I am therefore calling on the expert participants of this thread to share their view (-> @Bryan.Ford, @Oscar.Nierstrasz, @Fabrizio.Gilardi, @Bogdan.Warinschi, @David.Basin, @Vanessa.Teague, @Florian.Egloff, @Reto.Koenig, @Olivier.Pereira).

If I do not hear from you, I'm going to adopt tomorrow night.

Outside of these three proposals, there are also 3 additional proposals by @Bryan.Ford above, that did not get any feedback. That would be Ford-3, Ford-4 and Ford-5. See my comment above for a separate description of Bryan's proposals.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

All three proposals look good to me! (And no clear opinion right now about Ford-3-4-5.)

Marius Kobi @Marius.Kobi · 5 months ago Developer

Proposal 1 looks good to me.

Proposal 2: I am in favour of rejecting it. It seems to be based on an interpretation of the statements by the experts („find [...] implicit support“) that, in my view, is not fully supported by the statements.

Propsal 3 should be rejected. The orginal text reflects that the studies and the implementation of a PB are linked. The question whether the cantons should be required to implement a PB can only be answered after the studies have been completed. A PB should only be mandatory if it is widely seen as a meaningful tool and if the studies support its implementation. Propsoal 3 suggests that a decision on whether a PB has to be implemented has already been taken. That is, however, not the case. The summary is not the place for this discussion.

It is self-evident that cantonal particularities must be taken into consideration – that should be the case for almost every aspect of an e-voting system.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

I strongly support proposals 1 and 2. In particular proposal 2 nuances that there are tangible benefits from having a PB but with

additional trust assumptions and security specifics that need to be accounted for.

For 3 I don't have a strong opinion.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

From Ford-3,4,5 I understand @christian.folini 's doubts but also don't have strong opinions. Perhaps @Bryan.Ford can himself comment if he insists on his suggested changes?

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Some typos. 4B, para 2: "effect on the trust:" should be "effect on trust" 4B, last para: "The interface between the Public Board and these tools are" should be "The interface between the Public Board and these tools *is*" 4E, all paras: some text at the beginning of each para is missing

Collapse replies

Christian Folini @christian.folini · 4 months ago Maintainer

Good lord, whatever went wrong there with 4E! Glad you spotted that one and the other typos. Thank you.

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you everybody for chiming in or upvoting.

Proposal 1 (by @barbara.erni)

The proposal passes without opposition.

Proposal 2 (by @oliver.spycher)

The proposal has received two favorable comments and three upvotes from the experts vs. a negative comment with an upvote from two participants working for administrations on the canton level.

The proposal passes.

Proposal 3 (by @oliver.spycher)

The proposal has received a favorable comment and three upvotes from the experts vs. a negative comment with an upvote from two participants working for administrations on the canton level.

The proposal passes.

I have implemented all three changes.

All open issues within this discussion and the whole block 4 are thus resolved. I am thus closing this discussion.

Thank you very much for your repeated participation until the very end.

Edited by Christian Folini 4 months ago

Christian Folini @christian.folini closed 4 months ago

Christian Folini @christian.folini removed Last-Call label 4 months ago

Closed Opened 6 months ago by Christian Folini Reopen issue

Discussion 4B - Public Board as a means to gain trust (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

Is a PB suitable to establish trust?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 02, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

David Basin @David.Basin · 6 months ago Developer

A public bulletin board symbolizes something used in administrations that citizens are presumably comfortable with. It appears to enhance transparency too. All this is positive and can help establish trust.

If the entire system (including the BB) is not trustworthy and is hacked, then people will stop trusting it, despite the use of this nice symbol. So a central question is does this lead to a more secure design, potentially under weaker trust assumptions, than other alternatives.

Note that it may take some explaining and marketing for most users to understand there is a public BB used by the system and why they should trust the system more as a result. Presumably very few users (mainly experts and skeptics) will access the BB directly and most users will just react to what is displayed on their smartphone. From their perspective, it probably does not matter if the data comes from a public BB or a private database.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I agree that a BB is useful to establish trust, provided it is well-designed, properly-implemented, and used appropriately.

It is true that most people will not understand either the technical details or the threat model(s) of the BB, but will just think of it as a vague abstraction or metaphor. It is at this level that I think a "bulletin board", a "tamper-evident log", or a "blockchain" tend to represent almost-interchangeable abstractions in terms of the way most (non-expert and especially non-security-expert) people use them - even if the security/crypto experts can certainly engage in endless discussions about the fine distinctions between different definitions, designs, and threat models. It is up to the E-voting system designers, implementors, and validators to ensure that the BB chooses some reasonable threat model and design that is consistent both with the naive bulletin board metaphor and with the technical details of the way it's used in a voting system.

That is, I think it is reasonable to expect the existence and use of the BB to help gain public trust, inasmuch as the "public bulletin board" metaphor implies transparency and in particular the ability for "anyone" to inspect it for any misbehavior. But the technical details under the metaphor also of course need to stack up so that (a) it won't actually get compromised, undermining the public trust, and (b) independent experts who do pay attention to the underlying details won't find a lot of reasons to say it's insecure, not actually transparent, not actually a bulletin board, etc., which can certainly also undermine public trust regardless of whether it actually gets compromised.

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

I agree that in principle the PB could improve trust, since it increases transparency and it also gives more agency to interested users. But I also see the potential that it reduces trust, if voters do not understand the purposes of the PB, do not use it, or do not

use it correctly. I can also imagine that malicious actors might use the information available on the PB to create confusion about the accuracy of the results.

I seem to be the only one who is sceptical about the PB, so it's not unlikely that I'm missing something. Please tell me what I'm not seeing!

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

No worries Fabrizio, you are not the only one who is skeptical about this. And I think your comment above is right on target: In principle it sounds like a cool idea, but be careful to get it.

Florian Egloff @Florian.Egloff · 6 months ago Developer

Trust is something you can only build over time. The less experience you have with something, the less you should trust it. Could one build the system in a way that the "public board" becomes available when polls have closed? Voters in Switzerland are likely familiar with the idea of a public notice board for local referendum/election results. Thus, if one could build on existing trust structures, it may help to build trust in the overall system.

As @Fabrizio.Gilardi, I am sceptical that just "having" a public notice board would straightforwardly lead to "increased trust".

David Basin @David.Basin · 6 months ago Developer

I would think voters would like the idea that they or members of their party can check, e.g., that the tally was computed correctly. I can't see how this can reduce trust. Concerning confusion: it is necessary of course to do this in a way that disputes do not arise (as, for example, in the case of universal verifiability, everyone can check the tally and resolve the dispute themselves) or disputes can be unambiguously resolved in other ways (e.g. in the case of individual verifiability).

It is possible to have dispute resolution for all relevant aspects of voting, but it does require certain assumptions about the ability to communicate in a timely way and the protocols must, of course, support dispute resolution. See https://arxiv.org/abs/2005.03749 for more on this.

Collapse replies

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

> they or members of their party

That's a key difference. As I wrote earlier, I think the PB can be useful if the process is well structured, e.g. such that each party can nominate people to use the PB. Or what @Carsten.Schuermann suggests below (#21 (comment 687)), that is, some "representatives selected by the public" would be charged to use the PB. For example, a random sample of voters (to build on existing projects suggesting random selection as a representation tool) who are trained and supervised (e.g. by the Bundeskanzlei?) to do the job properly.

My bottom line is that setting up the PB and hoping that good things will follow is risky. In a more structured context, I definitely see the potential.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

An important property of a bulletin board is that it is independently verifiable, which means that first, it should be possible to verify that all entries on the board are valid and second that it is this board that is used when computing the final result. The board constitutes evidence, that all votes are valid, no vote has been copied, no one could have tampered with the votes, etc. As @David.Basin et al. wrote in 4A, monotonicity and final agreement are critical.

Making the board public from the get go seems to serve two purposes. It provides some kind of mechanism for individual verifiability *and* it sends a signal of transparency. Both purposes are important and create trust. If we look a little more closely, however, individual verifiability is at best partial, and typically, there is no evidence stored on the board that the vote was verified. (If there were, these entries would have to be verified again etc.)

If the bulletin board were not public and would only be revealed to auditors in the case of an audit, dispute resolution, or similar, it must still be independently verifiable, convincing auditors, scrutineers, or observers that the entries on the board reflect the voter's

intent. This could also be done in a way to create trust, but it is possibly more difficult than just making the bulletin board public in the first place.

I have to admit, I am still not sure if a public is a good idea or not. I used to be of the opinion, that nothing should leak from the election but the result, but in Internet Voting, I am growing increasingly convinced that it is also important to leak the evidence for why the result is the way it is, either to the public, or to representatives selected by the public.

Christian Folini @christian.folini · 5 months ago Maintainer

There is a separate attack vector mentioned by @Carsten.Schuermann in the discussion 4G. It is worth considering here, since it influences the trust in the PB.

Christian Folini @christian.folini mentioned in issue #26 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for contributing to this discussion. This is really fascinating. It's time to wrap it up, so here is my attempt at a summary:

Is a PB suitable to establish trust?

A Public Board can establish trust. If done right, the evidence and the transparency a Public Board provides will have a positive effect on the voter's confidence into the system, namely in a long term perspective. A Public Board signals the authorities' commitment to transparency, which in turn has the potential to translate into increased public confidence into the electoral process.

It is very important to note, however, that a misdesigned, a poorly implemented, a badly documented or an otherwise not optimally supported Public Board will have a negative effect on trust: Unless the Public Board and everything around it, is of a very high quality, it could very likely have a negative effect on trust. So the Public Board can only unleash its symbolic value if the simple metaphor is supported by good design, proper implementation and adequate operation that also convinces the experts.

It is likely that relatively few users would use the Public Board themselves, but instead rely on the checks executed by independent experts / political parties. But this is not necessarily a problem and one could think of establishing such a process officially, while still giving users access to the Public Board directly or via official and independent tools.

The Public Board is closely tied to the tools that make use of it, since very few voters will use it directly. The interface between the Public Board and these tools is therefore very important. If there is a failure with this interface, the Public Board will be broken from a user's perspective. That's why changes to the interface have to be done in a very careful way and the tools and their providers have to be supported adequately.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

[EDIT] Updated first paragraph based on proposal by Florian Egloff below. Also replaced one occurrence of "trust" with "voter's confidence". I have meant to perform this edit yesterday, but then I forgot to press the save button.

[EDIT] Two typos spotted in this summary after being quoted in 4A.

Edited by Christian Folini 4 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

I disagree with: "This works, because a Public Board is a powerful symbol or metaphor that is appealing to the voters."

If a PB can add additional trustworthiness, it is not due to its metaphorical nature, but due to adding the possibility of every voter to check the output of the system.

David Basin @David.Basin · 5 months ago Developer

I imagine people trust more things that they understand. The typical voter will not understand the details of consensus protocols, Merkel trees, cryptographic hash functions, and the like, even if they can check their votes using programs based on such mechanisms. Hence it is important to wrap these things in abstracter notions that they do understand, like that of a bulletin board.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your feedback @Florian.Egloff. This part of the summary is based on explanations by @Bryan.Ford above and I also read @David.Basin 's original comment as a similar statement.

But I agree, that the summary focuses too much on this idea. Let me rephrase that a bit:

- Replace with : "This works, because a Public Board allows the voters to examine the voting process themselves, which brings transparency. The Bulletin Board is thus a powerful symbol or metaphor that is appealing to the voters.
- Shorten "unleash its symbolic value" to "unleash its value"

Is this something you could agree with? If not, could you make a proposal that covers your position without throwing the metaphor completely out?

@David.Basin : Am I right, that you are OK with the wording of the summary?

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Maybe we could say that "A Public Board signals the authorities' commitment to transparency, which in turn translates into increased public confidence into the electoral processs." (I tend to avoid using the word trust, because it is too vague.) I much prefer "the voter's confidence" or "public confidence". Like @Florian.Egloff, I am also worried about using words like "symbols" or "metaphors" to describe trust-building. It is "verifiability" that ultimate builds public confidence.

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

I like that formulation because it moves away from the metaphorical nature but introduces the authorities' commitment to transparency. However, I would weaken the determinism from "translates" to "hast the potential to translate":

"A Public Board signals the authorities' commitment to transparency, which in turn has the potential to translate into increased public confidence into the electoral process."

Do you agree, @Carsten.Schuermann ?

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

@Florian.Egloff Sounds good.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you very much gentlemen. It's a pleasure to work with you.

I have edited the summary and implemented your new wording.

Christian Folini @christian.folini · 5 months ago Maintainer

We have updated a small part of the summary and it seems everybody agrees with the new wording (or the original summary that is very close to it).

So I am closing this discussion with the updated summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Christian Folini @christian.folini reopened 4 months ago

Christian Folini @christian.folini closed 4 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 4C - Public Board and user-friendliness (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

Does it seem likely that already or in the future a sufficient proportion of voters will use a correctly functioning device to perform the verification? How do you assess the potential user-friendliness?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 03, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

For me this is a critical point. From the description of the whole process, it is not entirely clear to me what possible scenarios might unfold from the point of view of the voter. If it is hard for me to follow, then I am sure it will not be easy for your average voter.

I see (at least) two issues:

1. convincing the user of the need for the PB and the verification process (the user must see a clear dded value), and
2. making the interaction clear, transparent, and efficient (i.e., "user-friendly").

I would like to see a classical usability study based on the various usage scenarios, using mock-ups of possible UIs and typical classes of voters as test subjects to assess the usability of what is envisioned, before investing in the full development of the actual system.

Collapse replies

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

This point is absolutely crucial. There is so much that normal voters (i.e. the vast majority) might not understand, or do wrong, with the PB.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

Supporting what @Oscar.Nierstrasz wrote, the question is not only if sufficiently many voters will do the verification, but also, if they do it attentively enough. Colleagues of mine did a study (https://fc20.ifca.ai/voting/papers/KVMR20.pdf), which also covers a Swiss Internet Voting system, with the result, that even if voters verify, only a fraction will notice errors. In the context of ballot marking devices, studies have shown that also here a large percentage of voters is attentive enough to spot manipulations.

Moritz Zaugg @moritz.zaugg · 6 months ago Developer

Usability is an important issue for the cantons. If a PB is not user-friendly, the voters won't use it. Wen and Buckland describe the problems of the vVote system (which was later decommissioned) in Australia (Victoria State) as follows "The system is highly

complex: for voters to use, for voters to understand, for election officials to supervise and assist voters, for effective auditing, and to be implemented and maintained with confidence." What do @Fabrizio.Gilardi, @Florian.Egloff and @Uwe.Serdult think about this issue?

Collapse replies

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

Yes, I completely agree that the PB introduces an additional layer of complexity into a system that is already complex.

As I said in another comment, maybe one way to proceed would be to embed the PB within a highly structured context, such as a citizen science project in which voters are trained and supervised by experts to use the PB on election day. But I don't know what's the minimal number if participants needed for the PB to make sense, and if that's practically feasible for a citizen science project.

Florian Egloff @Florian.Egloff · 5 months ago Developer

Sorry to be slow: Yes, usability is super important, as by definition it lowers the barrier to participate.

As stated elsewhere (#37 (comment 949)), I would couple the need for usability with the need for a clear strategy of how trust in the system can be formed. This goes beyond the "technical" analysis of whether the system is trustworthy, but has to include a political process for desigining&introducing such a technology.

One idea would be to work with communal councils (Gemeinderäte). Gemeindeversammlungen are great places to reach the interested voting public. One could start by building touchable systems: i.e. replicate the system with physical objects for a workshop-like experience. Give people the option to “touch” the system logic. Explain how trust in the system is established & give people the option to “practice” internet voting right there.

(I raised this originally in https://evote-dialog.okx.ch/federal-chancellery/evote-dialogue-questionnaire/-/blob/master/responses/experts/responses/individual-responses/5.5-egloff.txt).

David Basin @David.Basin · 6 months ago Developer

Moreover usability must be considered under the assumption that the adversary controls the user's interface on his/her laptop. For instance:

- Consider step 4a - Before sending the vote to the server, the encryption of the vote and the random number used for encrypting the vote are displayed on the laptop, here we assume one QR-code, it could also be two QR-codes (one per value). The voters scan the QR-code with their mobile. Based on the random number and the parameters downloaded in step 3, the mobile displays the encrypted voting options as text. Voters can also use additional devices with different software to verify the encryption repeatedly. They cast the vote if they are satisfied with the verification result.

In this step, rather than displaying the QR-code(s) to be scanned with the user's mobile phone, the laptop could simply print "Encryption Verified, please cast your vote".

- Alternatively, consider step 4b - Additional option: Prior to casting, voters have the option to create an arbitrary number of votes they do not intend to cast and verify the encryption of such a vote an arbitrary number of times. They can perform the verification with the same application on the same mobile or also with other devices possibly running different software. Once they are satisfied that the device they use for casting the vote encrypts votes correctly, they would create and cast the actual vote. It would be sufficient if the mobile only keeps the encryption of the vote, or in the case of two separate QR-codes, only to scan the one holding the encryption.

Here the laptop would not give the user this option and would simply report to the user "encryptions verification successfully checked, please create and cast your actual vote".

This is all consistent with trust assumptions that assume laptops are under adversarial control. In practice such attacks would probably fool some users. Others might notice the abuse and (I presume) would be told to go to the polling station to complete their vote.

The fact that humans are fallible has been studied in the context of security protocols. See e.g., http://people.inf.ethz.ch/basin/pubs/csf16-HumanErrors.pdf Many secure protocols are broken if humans make mistakes or can be tricked into making mistakes.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I agree that usability - both in general and of a bulletin-board verification mechanism in particular - is extremely important and needs to be studied and considered carefully. And I agree with @David.Basin that it's easy for subtle usability issues to translate into security/transparency weaknesses.

Nevertheless, we shouldn't let these issues distract from the fundamental value of having such transparency and verification mechanisms, even if only a fraction of voters might use them fully or correctly. One of the key reasons public transparency and verifiability is so powerful in a voting context is because the large total number of voters ensures that even if only a seemingly tiny fraction of them - e.g., 1% or less - actually use the verification mechanism [correctly], that will still typically leave an extremely high and perhaps overwhelming probability of misbehavior being detected - provided of course that other things don't go wrong like the attacker being able to predict in advance which voters will and which won't use the verification facilities.

So while both the usability and usable-security of transparency/verification mechanisms does need to be studied carefully and issues addressed, with the aim of ensuring that the fraction voters who will successfully use them will be as high as possible, nevertheless I don't think we need to expect or require that fraction to be all that high in order to conclude that the trouble and expense is worth the cost.

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

I agree on the value of the PB as a transparency tool. What worries me is that even though a small number of voters using the PB correctly might be enough for verification purposes, we need to think about the effects on trust if other users (possibly a larger number?) use it incorrectly. What happens if lots of people start posting alleged problems with the PB on social media? How quickly could those problems be fact checked? Does it even matter if those problems are not real if the doubts start spreading? Many people won't understand what the PB is anyways.

This scenario is not good for trust in the results, regardless of the actual performance of the PB as a verification tool.

Edited by Fabrizio Gilardi 6 months ago

Bryan Ford @Bryan.Ford · 6 months ago Developer

It seems like one of the security properties you'd want the PB to guarantee by design is that it should be infeasible for anyone to produce a "false proof of misbehavior". That's important, but I don't think it should be difficult to achieve in a reasonable, solid PB design.

Or are you envisioning other kinds of "alleged problems" that people might try to use to sow FUD about the PB? Can you suggest more specific examples of what you have in mind?

If there are availability/outage problems, for example, I can see the risk of people complaining about that - but I don't see how the PB would be different or make this problem worse in any obvious way than similar problems that might affect (say) users' ability to connect to the existing control components, the voting authority's website, etc.

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

I'm thinking about someone misreading or misinterpreting the information on the PB (maliciously or not) and then claiming on social media that the PB shows there was a problem with his/her vote.

Reto Koenig @Reto.Koenig · 5 months ago Developer

We would like to answer here for the original question:

> Does it seem likely that already or in the future a sufficient proportion of voters will use a correctly functioning device...

Our answer is a definite "**No**".

The reason behind the "No" is based on the following question: What exactly do you mean by 'correctly functioning device'. We are not targeting privacy here, but we are heading for correctness!

It cannot mean 'smartphone', for sure, as in practise, for most users, the smartphone and the Notebook / PC / Tablet are logically one device (they all share at least 'the same cloud'). Hence, in the large scale, we would not be allowed to 'trust' that the voters smartphone are functioning correctly, would we?

But if it does not mean 'smartphone' then what? Maybe using multiple smartphones? The smartphone of... ones spouse? But if so, then there would be no *user-friendliness*, would there? How would you want to explain that to all the fellow citizens?

So what is left? Well, the dedicated hardware device. But so far we have not seen such a dedicated hardware device in the Swiss context.

That is, why our answer is a definite: "No".

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

Let's say that the laptops of a significant proportion of voters are corrupted such that voters unknowingly cast manipulated votes. Clearly, the mobile will only allow a voter to detect the fraud if it was not corrupted in a manner corresponding with the laptop. @Reto.Koenig, if I get you correctly, you consider it to be quite certain that voters' mobiles are not useful for detecting the fraud given that their laptop is corrupted. I guess you agree that the chances of detecting the fraud would grow exponentially with every affected voter that performs the verification. But that only works if there is a chance that a mobile phone might work correctly despite a corrupted laptop.

This is an important point we need more clarity on and I would like to raise this question: Would a corrupted laptop in general imply that the owner's mobile is corrupted as well? Or is there reason to believe that the mobile could just as well still function correctly? Maybe we can relate to the Swiss systems where votes are cast with a browser, and let's assume that verification is performed with a mobile app.

Collapse replies

Sergio Alves Domingues @Sergio.Alves.Domingues · 5 months ago Developer

Assuming that a compromised laptop does - in most cases - not necessarily imply a compromised mobile device seems a reasonable assumption to me. Under such an assumption having a verification mechanism based on y mobile app could provide additional benefit in terms of security. Similar assumptions are in fact commonly used by systems like e-banking or more generally multi-factor authentication.

The whole PB mechanisms and the need to add yet other elements to the list of requirements for the voter seems however complex to me if it is meant to be adopted by a large portion of the voters. Even if only "partial" adoption is meant (which is my understanding), won't it have an impact on voter's trust, nevertheless? I'm thinking on concerns like "the system is so complex that it is not meant to be understood by everyone".

I however agree that it provides an additional transparency tool.

Reto Koenig @Reto.Koenig · 5 months ago Developer

> Would a corrupted laptop in general imply that the owner's mobile is corrupted as well?

No, it implies that at least one of the user's device has been successfully taken over by some adversary. But then a fair question arises: How did one figure out that any of which has been corrupted?

> Or is there reason to believe that the mobile could just as well still function correctly?

No, as some adversary already has taken over one device... why not the second one as well... maybe even prior to to the laptop? And yet again the question: How did one figure out that any of which has been corrupted?

We argue that even if there are two physical devices present, i.e. laptop, smartphone, logically they are fused to one single device, where **all** data is synchronized. The user is completely oblivious to where the data really is stored, the program really is executed. The user simply does not want to care about that. If one agrees on that, then how could one accept this convenience of transparency for the user, but denying the same convenience for the adversary?

A simple 'google' search for `smartphone zero day` or `smartphone zero click` presents quite a result... And it does not seem to remain on theoretical level though. So starting the attack via smartphone and then heading to the browser and only be active if browser and smartphone are under adversarial control seems absolutely reasonable to us.

The more 'trust-pressure' is put on the combination smartphone / browser, the stronger the incentive of its attack. So if Swiss democracy is at stake, the attack seems more and more worth it.

Talking of Google and Apple, the two vendors providing us with smartphone - software. What about their dominance in Browser - software?

Is it ok to call this 'independent'?

But in the end it all boils down to the one question: Does the combination of laptop / smartphone pose a possible angle of attack to the Swiss democracy, even for smaller budgets? In our opinion it is a simple: 'yes' without any 'but' attached.

Edited by Reto Koenig 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

I must respectfully disagree here with the highly-pessimistic stance that @Reto.Koenig expresses.

To address the most recent formulation of the question by @oliver.spycher directly - if a user's laptop is compromised does this mean that their mobile device is compromised as well? - my answer is definitely no: I think there is a rather high chance that the same user's mobile device will *not* have been compromised. And across a substantial number of users whose laptops might have been compromised (e.g., hundreds or thousands), it becomes extremely likely that at least a few of those users whose mobile devices will not have been compromised and who will detect vote manipulation by their laptop if they use cross-device verification correctly. There are two reasons I can think of to be reasonably optimistic here: mobile devices are quite different in both hardware and software from laptop/desktop devices, and mobile operating systems and hardware are traditionally more "locked down" and

secure by default against compromise.

The first reason boils down to the device diversity issue we've discussed at length before in the context of control components, but the diversity principle applies in general to the user-side devices just as much. At the moment, desktop/laptop-oriented operating systems like Windows and Mac OS are quite different in both hardware composition (ARM-dominated processors, highly-integrated chipsets specialized to handheld low-power devices versus Intel-dominated laptop processors and chipsets), in dominant operating systems (iOS and Andoid are quite different from Mac OS and Windows), and in application-level software ecosystems (laptops normally don't run iOS/Android apps and vice versa - they have to be built or manually ported and maintained independently). This diversity is far from complete or perfect of course: mobile and laptop/desktop OSs share many individual software libraries that may have common bugs, the different chips include standard hardware-design components that similarly may have common bugs, etc. But because mobile devices evolved around quite different usage models and requirements from laptop/desktop devices, the diversity between the mobile and laptop/desktop ecosystem is probably about the best one is likely to find in the overall computing device ecosystem today.

The other reason for optimism is simply because starting with its introduction of the iPhone, Apple set a precedent of "locking down" and securing mobile devices much more strongly - especially against applications that might behave maliciously unkbeknownst to the user - and other mobile devices (Android in particular) have followed suit, even if not quite as strongly. While laptop/desktop operating systems traditionally grant any application run by the user essentially the full power the user has, including the power to mess with other applications' files, to take over the screen or control other applications, etc., mobile operating systems generally at least try pretty hard (and iOS tries really hard) to protect applications from each other. This means that a voting or vote-verification app run on a mobile device generally has much stronger protection against being subverted by (say) a malware-infected application also installed by the user on the same device.

One must normally jailbreak an iOS device or otherwise find and exploit an extremely serious OS-level vulnerability in order for one application to control or subvert another's operation undetectably. Such exploits exist of course, and powerful nation-state adversaries in particular are likely to have such capabilities. But full iOS compromise exploits are now extremely rare and valuable, which means that even a nation-state attacker is likely to want to use such exploits in extremely carefully-targeted fashion in order to avoid the exploit being discovered, fixed, and hence "burned" or lost to the attacker. If the attacker does decide to use an exploit against a substantial number of mobile devices (to compromise many mobile verification devices in an election for example), the attacker faces a near-certain prospect that at least a few of those users will be expert enough to discover that something might be wrong and report the exploit or send the device to a forensics lab where it will be carefully analyzed and discovered. In fact, an untargeted attack attempt like this may face reasonable likelihood of accidentally finding and compromising at least one "honeypot" device run by one of the big tech or security companies, and hence being detected with near certainty. Thus, even an attacker with a powerful exploit will face an unattractive choice between using it to compromise only a few carefully-targeted users (hence affecting only a few votes), or to attack many less-targeted users but face the near-certainty that the exploit will be discovered quickly.

As mentioned above, Android has followed this "lockdown" precedent, though less strongly: Android devices also try to protect apps from each other, but OS-level compromises and jailbreaks have historically been somewhat easier to find against Android devices, especially because many hardware vendors have been lax about getting important security fixes passed from Google on to their devices' users in a timely fashion. But in this respect Android is still better than traditional desktop/laptop OSes have been, in that at least it's trying. (Desktop/laptop OSes are starting to catch up a bit in this respect, but they are hampered in this fashion by their need to support large existing desktop/laptop software ecosystems that would be broken by too much lockdown applied too quickly: this is why Apple hasn't really been able to mace the Mac App Store the "mandatory" way to install applications like it does on iOS.)

Note that even though Android devices tend to the device diversity issue, the diversity to be found between non-Apple laptops (typically Windows-based) and non-Apple mobile devices (typically Android-based) is extremely strong, because Windows and Android are completely different operating systems predominantly written in completely different languages (C/C++/C# versus Java) and most importantly, built largely from scratch and maintained by different and substantially-competing tech giants (Microsoft versus Google). This Windows-Android diversity thus may be seen as a significant benefit to laptop-mobile verification security in a voting system for typical non-Apple users, compensating somewhat for the somewhat weaker compromise protection a typical Android-based device might have. Apple users with a Mac OS laptop and an iOS mobile device will have somewhat less diversity between the two verification devices, but will benefit from the stronger "lockdown" measures in Apple's devices (especially the iOS devices but with Mac OS catching up in this respect too). Thus, considering diversity and OS-level protection in combination, there is fair reason to hope that both a typical "non-Apple user" and a typical "all-Apple user" will have fairly strong protection against both their laptop *and* mobile device being compromised at the same time.

Finally, @Reto.Koenig raises the question of how a user is to know whether a particular device is compromised? My answer to that question is, in general, the user doesn't - but that is irrelevant because the user shouldn't *need* to know that if the cross-device vote verification system is properly designed. If the laptop is compromised but the mobile device isn't, then vote manipulation by the laptop should be detected if the user follows the verification procedure correctly. Similarly, if the laptop is *uncompromised* but the mobile device *is* compromised, then vote manipulation should either be impossible (if it is the laptop casting the vote) or should still be detected (e.g., if the mobile device is casting the vote but the laptop is verifying it, assuming that process is supported). If one device is compromised and the other isn't, the user gets vote integrity protection regardless of whether the user knows (or ever learns) which device is or was compromised. It should only be the case in which an attacker has simultaneously compromised *both* the laptop and the mobile device that the attacker "wins" in manipulating the user's vote without detection. And based on the diversity and lockdown arguments above, each of these compromise risks is a separate, at-least-somewhat-independent probability. Across a large population of users whose laptops and mobile devices are simultaneously under attack, some of those users should with high probability have one or the other of their devices uncompromised and have a strong chance to detect the attack.

In short, provided the cross-device verification can be designed and implemented properly and made as usable as possible so that at least a reasonable fraction of users actually use it, I believe the current software/hardware ecosystem provides a fairly strong - and extremely useful from a security perspective - type of defense especially if verification is done between a laptop/desktop device and a mobile device.

Edited by Bryan Ford 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

I'd like to wrap up the discussion here in 4C. @oliver.spycher has tried to re-focus the discussion around the security of the device. If you want to chime in, then please try and do so tonight or Tuesday during the day. I'll post a draft summary Tuesday night (touching wood here. :))

Florian Egloff @Florian.Egloff mentioned in issue #37 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #20 (closed) 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your participation in this interesting thread. I have drafted a summary below.

@Bryan.Ford and the BFH people around @Reto.Koenig don't agree about the likelihood of voters that have a corrupted voting device (e.g. notebook) and verifying device (e.g. mobile). But I think this conflict can actually be avoided for the summary. Please check.

Does it seem likely that already or in the future a sufficient proportion of voters will use a correctly functioning device to perform the verification? How do you assess the potential user-friendliness?

It is very clear that user-friendliness is an important factor that decides on the degree of use by voters and also influences the proportion of voters that verify attentively enough to notice errors with the verification and also care to report them. However, requirements for user-friendliness in the context of internet voting must be specified before any thorough examination can be performed.

User interface studies are needed to assess various design and usage scenarios. It is also an area where testing of a design in a structured context like a citizen science project possibly in conjunction with e-voting trials could help studying the use of the public board and contribute back to its design.

Even if a mature and user-friendly design of the public board could be reached, there would still be the chance that an attacker could corrupt any particular voter's device(s) used for voting and / or verification. Likewise with social engineering attacks to steal secret information from the voter. It takes a sufficient proportion of voters that are non affected in order to detect large-scale fraud.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT] Replaced "the device used for voting and / or verification" with "any particular voter's device(s) used for voting and / or verification" on request of @Bryan.Ford .

[EDIT] Added a sentence on user-friendliness based on input by @Oscar.Nierstrasz below.

Edited by Christian Folini 5 months ago

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

I am missing language to the extent that even a mature "user-friendly" design would need a clear strategy of how such a system would be introduced and how trust in the system can be formed. I suggest adding it at the end: (I wrote the addition in *Italics*).

Even if a mature and user-friendly design of the public board could be reached, there would still be the chance that an attacker could corrupt any particular voter's device(s) used for voting and / or verification. Likewise with social engineering attacks to steal secret information from the voter. It takes a sufficient proportion of voters that are non affected in order to detect large-scale fraud. *In addition, even with a mature and user-friendly design, one would still need a strategy of how such a system would be introduced so as to enable citizens to understand and form trust in it.*

Christian Folini @christian.folini · 5 months ago Maintainer

That thought is missing, yes.

But is not this something that belongs into 4A, where the whole block 4 is being wrapped up?

I really do not want to avoid adding this and I think it is totally in line with what most experts have been saying here, but it could also be added to the communication discussion in 4D or the overarching 4A.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I'm fine with this summary. As a writing nitpick, you might change "...could corrupt the device used for voting and / or verification" to "...could corrupt any particular voter's device(s) used for voting and / or verification".

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bryan. Adopted.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I just noticed that the term "user-friendly" is not qualified. When I mentioned it above, I used it to cover "clear", "transparent" and "efficient". The term "user-friendly" means nothing in itself. Software quality standards qualify all non-functional "-ilities", and that must be done here too. It is necessary (at some point) to be precise about what parts of the the user experience must be optimized, and this will constitute the definition of "user-friendly" for the evoting project. For example, one might wish to (i) minimize user errors, (ii) minimize time required to learn how to use the system, (iii) detect user errors and recover from them in a given time, (iv) maximize the number of users who correctly follow up with the vote checking process. And so on.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for this comment. The term "user-friendliness" is already present in the question and I agree it is only used in a very casual way.

Here is a simple sentence that could be added to highlight the need for good base for the term:

> However, a definition of user-friendliness in the context of internet voting is needed before any thorough examination can be performed.

Is this enough to cover your concerns, or do you think a more elaborate wording is needed. If you prefer the latter, should I propose a wording or do you want to setup something yourself?

(FYI: The current plan is to add all summaries to the report and then have annexes with all the responses to the questionnaire as well as all the discussions here on the platform. So when looking at the summary, one would quite naturally find your more extensive argumentation in this regard.)

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Yes, let's keep it simple. I think "definition" is too absolute. I propose: "However, requirements for user-friendliness in the context of internet voting must be specified before any thorough examination can be performed." I gave some examples above and earlier. We could add these or leave them out, as you prefer.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Oscar. That is a good wording and I have adopted it in the summary.

I'd rather not bring the different examples in the summary; exactly to keep it brief and simple.

Christian Folini @christian.folini · 5 months ago Maintainer

Time to close this. Thank you for participating in this discussion.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 6 months ago by Christian Folini Reopen issue

Discussion 4D - Public Board communication (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

How should this additional method for verifiability be communicated towards the voters and brought into relation with the current method, knowing that up to now voters were told that only the electoral board is able to decrypt the votes?

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 04, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

Maybe it is just me, but am concerned about the proposed verification technique and to use the randomness to decrypt the vote. First of all, it is at odds with the general assumption that randomness is not shared for the encryption scheme to be secure (has indistinguishable encryptions, computation model). Second, an encryption scheme consists of three algorithms (Gen, Enc, Dec) and not four, i.e. including a second decryption algorithm Dec' that doesn't use the intended key but the randomness. Third, some voters will share photos of the barcode containing the randomness over social media, as a statement that they voted, not realizing, that anyone with access to the database of encrypted votes and the photo will be able to decrypt the vote (In some countries, it might even be illegal to use this technique, if vote privacy is not a right, but must be enforced). Fourth, meaningful verification can only take place, if the device used for voting is different from the device used for verification, meaning, no covet channels. But how can this be guaranteed convincingly, if laptop and phone are in bluetooth range or connected to the same home wifi network.

Regarding communication. If this verification technique should be used, then communicate to the voter that (1) two different devices should be used and both should have bluetooth turned off, and one should be connected to at the very least, a different network. Furthermore, it is important for the voter to know that the barcode should not be considered evidence that and how they voted, but it is a key to unlock the vote and should therefore not be copied, screen-shot, or photographed, but destroyed right after the vote is verified. The barcode is an essential part of the vote, designed to guarantee vote privacy. Don't lose it.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I understand the concern @Carsten.Schuermann points out that the current proposal makes it seem that the verification device uses the encrypted vote and the randomness to "decrypt" the vote, which indeed is often possible in practice but does not follow the formal APIs or analysis assumptions of typical encryption algorithms. Perhaps this gap could be addressed by clarifying what the verifying device is doing: e.g., not "decrypting" the encrypted vote per se but rather "replaying the encryption of the vote" (using the randomness necessary to perform that replay exactly) to confirm that the vote-casting device performed the encryption correctly. The inputs would then technically be the *cleartext* of the vote and the randomness, and the verifying device would be checking that it sees the exact same ciphertext output as the vote-casting device posts to the bulletin board.

Nevertheless, I'm not really happy with the current proposal in another more fundamental respect: namely the fact that the cleartext (or ciphertext plus randomness, which amounts to the same thing) needs to be "exposed" completely to the verifying device at all. This means effectively that the verification threat model is:

- For vote integrity protection, the user trusts that *at least one* of their devices is uncompromised and can thus detect vote manipulation.
- But for vote privacy protection, the user must trust that *all* of their devices are uncompromised, since all verifying devices as well as the vote-casting device will know the cleartext vote and can leak it to any adversary.

While I admit that reasonable arguments can be made that this threat model, protecting integrity over privacy, represents an understandable and perhaps tolerable balance, I don't think it should be necessary for the privacy-protection threat model to be this weak. In particular, can we design a verification mechanism that requires *only* the vote-casting device to be uncompromised in order to protect the voter's privacy? I think we can.

An alternative proposal:

Suppose that the multi-device verification process works a bit more like an electronically-automated variation of the code-checking process currently provided by the mailed voting cards.

In particular, once the voter prepares the vote on their (trusted for privacy) vote-casting device, they engage in some simple interactive process with their verifying device(s) - either by scanning QR codes or over the network - such that the verifying device shows an ephemeral pseudo-random code for each choice of candidate, without ever knowing or needing to learn which candidate the user actually chose. The vote-casting device at the same time shows the code corresponding to the candidate actually chosen, which should match the code displayed by the verifying device for that candidate. The user is thus responsible for ensuring that the code displayed on the vote-casting device matches up with "the correct one" of the codes displayed on the verification device, without ever needing to give that information to the verification device.

For this to be secure, the vote-casting device must of course be unable to learn the codes for any candidates other than the one for which the vote is encrypted. This could be done with an oblivious transfer protocol similar to the way the current paper-based code system works, for example, or in many other ways. Since the codes can all be ephemeral and unique to a particular verification interaction, this process can be done repeatedly with multiple verification devices, for multiple potential votes before the user actually casts the ballot, and perhaps can be done again with multiple verification devices *after* the user casts the ballot as well.

The usability question, of course, is whether the need for the user to match up codes instead of seeing "You voted X" on the verification device will be a significant deterrent, making too many users not do verification at all (or correctly). But given that users of the current system are already expected to check codes against paper cards, being asked to check (different) codes against the displays of verification devices in an electronic verification process does not seem to add too much to the learning curve, and would at least offer some consistency.

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

To me, this discussion shows how hard it would be to make sure that voters understand and use the PB correctly.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

@Bryan.Ford If the device is designed to replay the encryption, then encryptions are deterministic, which means that the crypto system is not IND-CPA secure. This opens another can of worms...

@Fabrizio.Gilardi It is not just how to use the PB, the question is also what guarantees individual verifiability can give, and under what what additional assumptions.

Christian Folini @christian.folini changed the description 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for contributing to this discussion. It's time to wrap it up, so here is my attempt at a summary.

How should this additional method for verifiability be communicated towards the voters and brought into relation with the current method, knowing that up to now voters were told that only the electoral board is able to decrypt the votes?

Effective communication with the voters is important. This is particularly true with regards to good practice when performing the verification (e.g. different device, separate network, bluetooth turned off) but also with regards to possible security risks that emerge from offering a new means of verifiability.

It would be useful if a scheme could be developed that does not require to display information (e.g. randomness) that is critical for the secrecy of the vote.

The difficulty to communicate the proper use of a public board effectively, makes it a hard to solve problem.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 4E - Public Board and voting secrecy (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

How likely does it seem that in the far future decrypted votes will be correlated with the voters' identities? Would we need to consider this a problem from societal perspective (as by now voting secrecy is guaranteed by swiss law)?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 05, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Reto Koenig @Reto.Koenig · 6 months ago Developer

This definitely depends on the protocol in use. For the actual protocols in use based on mix-net or homomorphic-tallying, it will be overwhelmingly likely (100%).

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

Agree with @Reto.Koenig . But of course (as @Olivier.Pereira pointed out in his answer to the questionnaire) there are other protocols that post commitments to the BB instead, which are information-theoretically secure - the protocol property is called 'everlasting privacy.' See for example Moran and Naor: https://www.talmoran.net/papers/MN06-voting.pdf For these protocols, it is not possible for a future attacker to decrypt votes.

Edited by Vanessa Teague 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

I agree with your assessment of the likeliness of the votes being decrypted, but how would you correlate the clear text vote with the identity of the individual voter?

Do we have to assume entities exist that save dynamic IP address assignment for potential future decryption of votes?

Maybe @Adrian.Perrig wants to chime in here as well.

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

I think that an important risk comes from the voters themselves: they will have a vote "fingerprint" (a hash, ...) in their hands in order to verify that their vote appears on the bulletin board. This is an information that voters will have to keep stored or written somewhere (it will be too long to remember it), and it is then difficult to expect that it will remain secret. (There may even be reasons to make it publicly available to external auditors.) The link between the voter and his vote on the board is then disclosed.

Christian Folini @christian.folini · 6 months ago Maintainer

I see that angle, Olivier. But I doubt it scales. There are people posting their voting material to facebook. This is not forbidden in Switzerland, AFAIK.

And if it does not scale, then you are targeting individual voters and then it's probably easier to steal it from the browser instead of waiting for future decryption.

I am not saying that the problem of limited duration of the secret should be ignored. I am arguing, that a technique should not be rejected over a long-term problem when short-term problems are accepted.

From a threat modeling perspective, you could also ask who has an interest in breaking voting secrecy of former votes, dating several years back. I can not see a non-domestic player having an interest in this (maybe with the exception of Swiss citizens living abroad in an oppressed state where their Swiss votes might be interesting in rare occasions).

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

@christian.folini : Independent of everlasting privacy schemes on public BBs, the EMB will be able to break vote privacy in some years time, as long as they kept a backup database relating voter identity and encrypted vote. We would need to trust the EMB that they will not do that.

@Olivier.Pereira correct me if I am wrong, but I always thought that everlasting privacy schemes require the exchange of some secrets between the participating parties ahead of time. If this exchange is compromised then so is the entire scheme?

Collapse replies

Bogdan Warinschi @Bogdan.Warinschi · 6 months ago Developer

Everlasting privacy schemes require channels between voters and the tallying authorities. As far as I know, the security models assumed for everlasting privacy consider an unbounded adversary but one who is privy only to the public output of the protocol and cannot interfere/observe the voter channels to the authorities.

Perhaps here, the question is also one of scale: how much information would an adversary need to collect so that later, when it can invalidate some hardness assumption and break say TLS, it can deanonymize users at scale.

NB: @christian.folini is the relevant question here whether the PB somehow enables an attack that scales or if it can be used to deanonymize some specific user?

Christian Folini @christian.folini · 6 months ago Maintainer

If you ask me, I think it does matter. If I want to target a specific voter, then there seem to be several attack vectors at my disposal, some of them viable for the physical voting channels too.

But if a BB allows to retrieve unencrypted votes and match them to the identities of the voters within a couple of years, then I think it's a different ballgame.

Personally, I also think the legislator ought to prioritize on CIA (Confidentiality, Integrity and Availability). The law does not really align with these terms. But given the federal court allows Landsgemeinden, I think there is in fact an implicit priority on the integrity of the vote over the confidentiality. That could conceptually open a gap where you could allow the adoption of a voting scheme that does not fulfill everlasting privacy requirements. But I take it, that's an unpopular opinion for multiple reasons.

Christian Folini @christian.folini · 6 months ago Maintainer

@Carsten.Schuermann : I totally agree on the attack vector stemming from the Electoral Management Body.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

One remark regarding scale. Formal models and proofs usually target qualitative results (a system is secure or not) but are quite bad at providing quantitative results. I would also exepct that the approaches which try to provide quantitative guarantees (e.g. concrete security reductions) would not work well in the context of evoting schemes. To sum up, given the state of the art, it may be difficult to provide formal guarantees that attacks that scale are not posible.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I agree that the votes encrypted today may well be decryptable eventually (using future cryptographic breakthroughs and/or quantum computers), and that this is an important problem that needs to be considered and ideally addressed if possible.

Further, unless the BB design has at least post-quantum if not information-theoretic anonymity mechanisms built into the posting and/or reading mechanisms, there is a real risk that it will be possible for powerful (state-level) adversaries to correlate posted votes with voter identities, through traffic analysis of recorded network logs and the like. We don't need to assume the interested adversary itself has necessarily observed and saved in long-term storage all the information necessary to perform this kind of network-level linking, although some potential state-level adversaries are certainly capable of and probably already storing the data

necessary to do this. Besides states, many tech companies including tech giants are collecting and recording all the traffic analysis and user-identity-correlation inference data they can for advertising and marketing purposes. Many potential adversaries who might not have all this data when it is first recorded might likely have come by it (through direct hacks or third-party dumps, etc) in the more distant future once vote ciphertexts become decryptable and it becomes interesting to them to associate encrypted votes with voter identities.

I agree that there are reasonable and interesting solutions to this long-term privacy problem that are worth exploring. One is the use of only information-theoretically private commitments and proofs on the public BB, as @Vanessa.Teague suggested. Another approach would be to use information-theoretic and/or post-quantum crypto techniques to anonymize user posts to and reads from the BB. There are many interesting potential ways to do this, but it would certainly add some complexity to (or around) the BB design.

Social implications of long-term privacy risks

But let's return to the question of what this long-term privacy risk means from a societal perspective. What are the realistic motivations of an adversary who might want to decrypt a long-past vote, correlate it with the voter's identity, and act on that information in some way? The two main motivations I can think of are closely-related, basically embarrassment or coercion. I see it as potentially realistic that both Swiss and non-Swiss adversaries might want to hurt some political candidate's campaign, or "help" their rival's campaign, by revealing how the targeted candidate voted in the past. This is just a form of the classic "digging up dirt from a candidate's political history" attack - and the candidate need not have even had the vague intent of getting into politics at the time they cast the vote that later gets decrypted and used against them.

But this form of embarrassment risk seems in general likely to fade with time. It's hard to imagine how the party or propositions a future political candidate voted for or against 10 or 20 years ago will likely be *more* embarrassing for this purpose than the "usual dirt" that tends to get dug up and used anyway, e.g., based on what they said 10-20 years ago, or what other people report that they said or did then, or the drunken beer parties they're known to have attended in college or whatnot. So frankly I don't currently see how this risk of embarrassment about long-past votes is necessarily a big enough issue by itself to warrant a lot of additional cost or complexity in an E-voting system.

The other obvious potential motivation I see is coercion: the voter promises to be good and vote as they're instructed, and the coercer offers a reward for doing so and/or a promise to come break an arm or leg if it's (ever) discovered that the voter reneged on that promise. But then the current Swiss E-voting system does not even attempt to address coercion, so although as I've stated I think it *should* be considered a risk to take seriously, in the current threat model a coercer doesn't even need to wait for cryptographic breakthroughs or quantum computers to succeed at this attack.

So in summary, while I definitely agree that the long-term vote privacy problem is an important problem to consider carefully, I'm not convinced (yet) that it's critical enough to warrant a lot of design cost and complexity - especially if it provides only a weak form of long-term "privacy" that addresses only vote-embarrassment risks but not vote-coercion risks.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I also would like to propose a terminology distinction between "long-term privacy" and "everlasting privacy" as potential goals to aim for.

To me, "everlasting privacy" means "information-theoretic privacy", which I think is an unrealistic goal or expectation to set in its pure form. As @Bogdan.Warinschi pointed out, everlasting privacy schemes always in practice assume that some kind of abstract "private channel" exists between voters and election authority or BB components. The only practical way to satisfy these "private channel" assumptions in the real world, today, is using encrypted communication protocols such as TLS. No TLS or other practical encrypted communication protocol I'm aware of can guarantee true "everlasting privacy" in whatever content flows through the encrypted channel.

To me, "long-term privacy" sets the slightly-weaker but more-realistic expectation of privacy we have reason to believe will hold up against either forseeable improvements in cryptanalysis and/or quantum computers. It's hard to judge the first factor, of course: in principle some mathematical or cryptanalytic breakthrough could blow away any or all cryptographic algorithms in existence at any moment, but I'm pretty sure everyone is largely agreed that this is extremely unlikely. The risks that seem more likely and readily forseeable are either (a) scalable quantum computers arrive and render all of our discrete-log and factoring-based public-key cryptosystems insecure while probably leaving the "post-quantum" crypto systems (e.g., lattice cryptography) largely intact, or (b) new cryptanalytic discoveries render the *less-mature* and *less-stable* schemes currently in use - such as pairing-based and lattice cryptosystems - much less secure than we'd hoped. There have been significant recent weakenings in popular pairing-based curves, for example, and a lot of questions about which lattice crypto systems and other post-quantum schemes will stand the test of time, just because they're so relatively new and haven't yet been subject to vast amounts of scrutiny.

There are now concerted, ongoing efforts by major companies and standards bodies to bring "post-quantum" encryption at least to TLS: see for example NIST's post-quantum cryptography project, Microsoft Research's, CloudFlare's, Google's, and Amazon's. So we have reasonable expectation of seeing widely-supported and standardized post-quantum TLS communication channels becoming available soon.

Further, the standardization efforts for post-quantum TLS are attempting to design conservatively to achieve "best-of-either" security and privacy between a mature classic (but non-quantum-safe) key exchange algorithm and a less-mature but apparently quantum-safe key exchange algorithm. Thus, these protocols "should" remain secure for as long as one of the two underlying schemes holds up against attack, whichever is longer. In other words, these schemes "should" survive one of (a) the arrival of scalable quantum computers, or (b) a successful, classical cryptanalytic attack against the new post-quantum crypto systems. But these proposed post-quantum TLS schemes won't survive both events (a) and (b) in combinations; that's the significant remaining risk. I hope that the probability of this combination-compromise event happening in the forseeable future should be fairly small;

perhaps the hard-core cryptographers can comment on their perspectives.

At any rate, if we believe that these post-quantum TLS efforts will succeed and hold their security for a fairly long time as they're intended, then I suggest we consider this a reasonable and practical basis for achieving what I would term at least "long-term privacy" if not "everlasting privacy". By designing a public bulletin board to achieve information-theoretic or "everlasting" privacy under the practically-unrealistic assumption of perfect private channels, then replacing the perfect private channels with post-quantum TLS communication channels would hopefully give the bulletin board design what we could at least call "long-term privacy".

Edited by Bryan Ford 6 months ago

Olivier Pereira @Olivier.Pereira · 6 months ago Developer

Hi,

I think that there is another very important distinction in this everlasting privacy question, that is, everlasting privacy towards whom.

Clearly (as pointed by @Bogdan.Warinschi and @Carsten.Schuermann), if we want everlasting privacy towards the election authorities and/or people monitoring the network, we need some extra assumptions (typically on channels) that probably can't be realized at scale in practice. The reason for the need of these assumptions is that, in the information theoretic sense, the information on the votes must travel from the voters to the talliers. Otherwise, tally is just impossible. And if the information travels, and if the adversary has access to all channels, then this is just incompatible with everlasting privacy.

I believe that a more interesting and practical goal is to have everlasting privacy for the bulletin board content. The intent here would be to guarantee that the addition of a public bulletin board does not create any *new* privacy risk.

We have all the privacy risks that are there in the current system w.r.t. the election authorities, the ISP, ... and the goal would not be to touch these.

But we are discussing the addition of a bulletin board, which essentially makes information conveniently available to just everyone. Here, there is a possibility to make sure that the bulletin board does not leak any single bit of information about the votes, in the information theoretic sense of everlasting privacy.

This can be achieved just with our usual communication channels (which should answer a question by @Carsten.Schuermann). For instance, the voter can send to the authorities an encryption of their vote, and the authorities would only post on the bulletin board a perfectly hiding commitment on that same vote, while making sure that the voter can be convinced that this is indeed a commitment to his vote.

Edited by Olivier Pereira 6 months ago

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Olivier. I am not sure if there is not a negation missing in the following sentence of your statement:

> And if the information travels, and if the adversary has access to all channels, then this is just compatible with everlasting privacy.

If yes, you may want to edit it.

Bryan Ford @Bryan.Ford mentioned in issue #25 (closed) 6 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

1. How likely does it seem that in the far future decrypted votes will be correlated with the voters' identities? I refer to the other experts to answer this.
2. Would we need to consider this a problem from societal perspective (as by now voting secrecy is guaranteed by swiss law)? Yes, we would need to consider this a problem. Voting secrecy is guaranteed for lots of reasons and constitutes a basic democratic right. If the platform cannot offer that, it should not be introduced. It goes even further: if the platform introduces doubts around secrecy, this perception of "non-secrecy" can itself influence the vote. This alone should be a reason to adopt a healthy dose of scepticism towards introducing public elements without a clear & convincing concept of how voters will interact with it.

See e.g.:

1. https://www.cambridge.org/core/journals/british-journal-of-political-science/article/is-there-a-secret-ballot-ballot-secrecy-

perceptions-and-their-implications-for-voting-behaviour/CD20CED9A450927354E8E754FC67D5E3

2. https://onlinelibrary.wiley.com/doi/full/10.1111/ajps.12019?casa_token=NN_KA8p67JIAAAAA%3AFpyCxhkwSZYtfsW7VxnJygaQjP5sMt1j4_zedd45lI2DhAJ_UM7eFAIMxZwf3igSOy1Tj322FXoj5Wv-

Christian Folini @christian.folini · 5 months ago

Maintainer

Thank you for contributing to this discussion. It's time to wrap it up, so here is my attempt at a summary.

I have read through your responses carefully and the line of thought I see is that publishing encrypted votes is risky and securing that is depending on technology that does not exist today (See Bryan's explanations on improving TLS). The alternative are the publication of perfect hiding commitments instead of the encrypted votes. That technology exists today. So I am focusing on that design approach in the summary and cover the alternative only briefly. If my interpretation is too simplistic, then please shout.

Summary:

How likely does it seem that in the far future decrypted votes will be correlated with the voters' identities? Would we need to consider this a problem from societal perspective (as by now voting secrecy is guaranteed by Swiss law)?

Voting secrecy is a basic right, guaranteed by law. If a Public Board raises doubts about this guarantee, then this undermines not only trust in the Public Board but also in the internet voting channel and voting as a whole. The use of a Public Board raises some concerns regarding vote privacy. Depending on the cryptographic techniques used, advances in cryptanalysis, or due to bugs, the information on the board may reveal how a voter voted.

We can only speculate on the potential motivations for future attackers to attempt decryption or de-anonymization of past votes, and what harms such attacks could lead to, such as voter embarrassment or coercion. Nevertheless, it is important to understand the risks and trade-offs associated to using a Public Board, and different designs thereof.

Some Public Board designs publish encryption of votes. Here, secrecy of votes relies on the same set of assumptions which guarantee secret communication over the internet. Yet, these assumptions may be invalidated, and votes may be decrypted if quantum computers become a reality. If the board contains voter identifying information (e.g. required for auditing purposes) then the link between voters and their vote may be revealed. These designs may potentially be strengthened by using encryption schemes which are secure even against quantum computers -- such schemes are under development by the cryptographic community.

Other designs may hide the link between voters and their encrypted ballots so even if ballots get decrypted individual choices stay secret.

Finally, other designs aim to achieve "everlasting privacy". Instead of publishing encryptions of votes, such schemes publish only a so-called "perfectly hiding commitment", which registers the vote on the bulletin board, analogous to a hash of the encrypted vote, but which provably contains no information about the vote's content that could ever be decrypted or revealed no matter how powerful the attacker is. Of course, the overall guarantees for vote privacy still rely on the security of the rest of the building blocks.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

[EDIT] Typo
[EDIT] Replaced original summary with the rewrite from @Bogdan.Warinschi below.
[EDIT] Replaced *"While one can question the benefit of this information for these actors in the possibly distant future and the benefits look rather limited in that light (both for embarrassment and coercion of the voters) it is important to understand the risks and trade-offs associated to using a Public Board, and different designs thereof."* with *"We can only speculate on the potential motivations for future attackers to attempt decryption or de-anonymization of past votes, and what harms such attacks could lead to, such as voter embarrassment or coercion. Nevertheless, it is important to understand the risks and trade-offs associated to using a Public Board, and different designs thereof."* on request of @Bryan.Ford .
[EDIT] Replaced *"Finally, other designs aim to achieve "everlasting privacy": instead of publishing encryptions of votes, such schemes publish a so called "perfectly hiding commitment" -- essentially a hash of the encrypted vote. By themselves, such Public Boards reveal absolutely no information about the votes. "* with *"Finally, other designs aim to achieve "everlasting privacy". Instead of publishing encryptions of votes, such schemes publish only a so-called "perfectly hiding commitment", which registers the vote on the bulletin board, analogous to a hash of the encrypted vote, but which provably contains no information about the vote's content that could ever be decrypted or revealed no matter how powerful the attacker is."* on request of @Bryan.Ford .

Edited by Christian Folini 5 months ago

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago

Developer

I would rephrase:

> While one can question the benefit of this information for these actors in the possibly distant future and the benefits look rather limited in that light (both for embarrassment and coercion of the voters) it is important to understand the risks and trade-offs associated to using a Pubic Board, and different designs thereof.

To: (Reason: The benefits in the distant future are highly contextual and remain speculative.) "While one can question the benefit of deanonymization in the distant future, it is important to understand the risks and trade-offs associated with using a Public Board and different designs thereof."

In addition: I would also add a conclusion statement that assesses these three versions. I would strongly oppose a PB that follows the first type of design, and I suspect, many experts on this dialogue would agree. For me, the system has to de-link the vote from the individual. We must ensure that a catastrophic loss of privacy is prevented by design to the best extent possible.

Christian Folini @christian.folini · 5 months ago Maintainer

It is speculative, but it has been part of this discussion nevertheless and I have not heard other arguments. Yet, I am OK with dropping that part of the sentence, if there is no objection.

@Bogdan.Warinschi : Do you agree?

@Bryan.Ford : You brought this up? Are you insisting it's in the summary. If I do not hear from you, I assume it's OK.

As for the addition: How about this? "Regardless of the design used for the Public Board, the system has to guarantee the vote can not be linked to an individual to the best extent possible."

Edited by Christian Folini 5 months ago

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Two quick reactions:

- Reading this again, the "possibly distant future" wording may be confusing, indeed: I am not sure of what it adds -- it may or may not be in a distant future. I do not know what is more likely between DDH being broken at the 2048 bits level in the next 20 years and a bug in a PRG (or any other bug in a protocol implementation) that immediately leaks the votes displayed on the board. My guess would be the latter. (Of course, it seems better to decrease the number of assumptions on which we rely: it seems better to have a system that relies only on a good PRG than a system that relies on a good PRG and on DDH.)
- When writing: "the system has to guarantee the vote can not be linked to an individual to the best extent possible" -- does this refer to the encrypted vote, or to the vote in clear? If it is to the encrypted vote, I would rather think that it is most important, for auditing purpose, to make sure that the system can link a voter and his encrypted vote, even if this link is not published on the board -- this seems to be a central safeguard against ballot stuffing. Besides, if the system operator is corrupted, he will anyway be able to make that link. As a result, it seems much better to require the link to exist, so that it can support auditing, and not just be helpful to a cheating operator.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

@christian.folini I'm ok with dropping the speculation about when/how/to whom the information may be useful. The key point is that there are trade-offs behind each design and that it is important to explicitly identify and accept those that come with the adopted design(s).

@Olivier.Pereira Regarding the link between individual voters and their (encrypted) ballots: it seems that being able to link voters with (encrypted) ballots *publicly* is what causes the issues. Bad PRGs or leaked decryption keys would allow deanonymization at scale. One can imagine that the linking information needed for auditing can be stored/used from an additional trust domain so that one can get the best of both worlds, no?

Christian Folini @christian.folini · 5 months ago Maintainer

I'm waiting for Bryan to chime in with regards to the speculation that was criticized.

For the rest of the summary, I see a need that @Olivier.Pereira and @Bogdan.Warinschi reconcile their positions. If possible.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

@Bogdan.Warinschi yes, I totally agree! I wasn't sure that the sentence "the system has to guarantee the vote can not be linked to an individual to the best extent possible" would not be interpreted as "we all agree that it would be good to have a protocol that completely erases the link between voters and their encrypted vote". Removing this link from the board looks good for defense in-depth (even though it complicates verifiability), but the link also seems very useful for auditing purpose, so we should keep it *somehow*.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I would also be ok with dropping the "While one can question the benefit of deanonymization in the distant future," entirely. I just wanted to remove the assessment that such benefits look rather limited, as it is contextual to what actor you imagine to encounter

– the more evil you expect the future adversary to be, the more concerning access to true (believed to be secret) information on people's preferences becomes. I would find @Olivier.Pereira 's explanation that it may even be a reality in the near-term even more concerning.

re the linkability: yes, my concern was that catastrophic failure of the PB should not immediately allow linking voters with ballots. Sorry if I was not precise enough.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

@christian.folini I think me and Olivier are in agreement. :)

Bryan Ford @Bryan.Ford · 5 months ago Developer

May I suggest rephrasing the "While one can question..." part more-or-less as follows?

> We can only speculate on the potential motivations for future attackers to attempt decryption or de-anonymization of past votes, and what harms such attacks could lead to, such as voter embarrassment or coercion. Nevertheless, it is important to understand the risks and trade-offs associated to using a Pubic Board, and different designs thereof

I think this phrasing clearly labels the speculation as such, while I think usefully, offers the reader at least a couple concrete (if speculative) examples to think about.

Florian Egloff @Florian.Egloff · 5 months ago Developer

looks great to me

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bryan. I have edited the summary and adopted your wording as it keeps the original argument but makes the speculation more transparent.

@Florian.Egloff apparently agrees. Thank you Florian.

@Bogdan.Warinschi : You also commented on that paragraph. Are you OK with the new wording of said paragraph? If I do not hear from you, I assume consent.

Edited by Christian Folini 5 months ago

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

All good with me -- the paragraph is much more accurate.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bogdan.

Florian Egloff @Florian.Egloff · 5 months ago Developer

The summary seems fine to me. the only question I have is: what measures will be adopted that the introduction of internet voting does NOT raise doubts with regard to the secrecy of the vote? possibly our discussion over at #37 (closed) (re the participation of the public) can shed some light on this?

Christian Folini @christian.folini · 5 months ago Maintainer

Are you referring to the Public Board, or internet voting as a whole?

Florian Egloff @Florian.Egloff · 5 months ago Developer

The risk of introducing doubts regarding the secrecy of the vote applies to both and would need to be addressed. Note this is again one of those technical / social splits: you can technically be more secure to guarantee the secrecy of the vote than the current system allows for, but socially introduce doubts regarding that new system. I would suggest any election authority introducing such a new channel would need a clear strategy of how you prevent that from happening, which would depend at the very least on having a clear mental model of which actions will contribute to voters being able to form trust in the new technology. this is also to mitigate the larger risks of "turning away" from a technology that @Bryan.Ford raised in #19 (closed).

Christian Folini @christian.folini · 5 months ago Maintainer

I think reflections as this are best suited for the *Big Picture* discussion.

Florian Egloff @Florian.Egloff · 5 months ago Developer

great. Would you mind opening the Big Picture discussion?

Christian Folini @christian.folini · 5 months ago — Maintainer

Almost there. Likely tomorrow.

Christian Folini @christian.folini added Last-Call label 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago — Developer

just a typo in: "A possible architecture that achieves this characteristic is publishing no the encrypted vote,"

"no" should be "only"...

Collapse replies

Christian Folini @christian.folini · 5 months ago — Maintainer

Thank you. Fixed.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago — Developer

> A possible architecture that achieves this characteristic is publishing only the encrypted vote, but a so-called "perfectly hiding commitment" - essentially a hash of the encrypted vote.

This doesn't sound correct. You probably want something along the lines: A possible architecture with this feature does not publish encrypted votes. Instead, for each vote it publishes a so-called "perfectly hiding commitment" -- essentially a hash of the encrypted vote.

NB: I think equating a perfectly hiding commitment to a hash may be somewhat misleading but it's not entirely wrong. I'm wondering what is it buying.

Christian Folini @christian.folini · 5 months ago — Maintainer

@Bogdan.Warinschi : Yes, I was too quick. Should have noticed this yesterday. I have now rewritten this to *A possible architecture with this feature does not publish encrypted votes*.

@Florian.Egloff: Do you agree with the sentence as it is now or do you really understand it as *only*.

If that's out of the way, then let me explain why I am equating the "perfectly hiding commitment" with a hash. I assume a hash is something that people reading our summary are familiar with. And if not, it is relatively easy to explain how a it identifies something, without revealing any privacy relevant information. The situation is different with the "perfectly hiding commitment", which is a very technical term, that is harder to explain.

Bogdan: Do you think it is correct enough to stay that way, or should we try and find a different wording?

Florian Egloff @Florian.Egloff · 5 months ago — Developer

ignore my comment re "only". I just wanted to let you know there is a typo. should not have suggested what you meant...

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago — Developer

@christian.folini I assumed that something along the lines above is what motivated using a hash as analogy. Technically, this is not accurate though, and just hashing a message does not ensure that the message is hidden (image that the hash function reveals the first few bits of the message -- this doesn't break any of the typical properties of a cryptographic hash function).

Having said that, I'm happy to deffer judgement on this: it's sometimes better to improve the information at the expense of being precise.

Bryan Ford @Bryan.Ford · 5 months ago — Developer

The technical inaccuracy of implying that a perfectly hiding commitment is "essentially a hash of the encrypted vote" also bothers me, and I think the wording can be fixed so as not to say or imply this while still allowing the analogy for clarity. How about this:

> Finally, other designs aim to achieve "everlasting privacy". Instead of publishing encryptions of votes, such schemes publish only a so-called "perfectly hiding commitment", which registers the vote on the bulletin board, analogous to a hash of the encrypted vote, but which provably contains no information about the vote's content that could ever be decrypted or revealed no matter how powerful the attacker is.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Sorry to bring this up so late in the discussion, but it was triggered by @christian.folini 's summary:

I think the summary is a bit too black-or-white and restricts the design space. The text implicitly talks about a powerful adversary, but then suggests a design intended for (essentially) unbounded adversaries (information-theoretic security). NB: Cryptographers would be more worried about a break of assumptions due to quantum computing. This can be dealt with using quantum-safe primitives, and there has already been some progress in that direction (e.g. [https://eprint.iacr.org/2017/1235]).

I like @Bryan.Ford 's suggestion to try to be more nuanced. I would go one step further and still leave room for designs which do not meet the long-term privacy bar. If one casts security of e-voting in a broader context: with current designs vote privacy would break down together with the entire internet infrastructure. Furthermore, with voting one can be much more agile than with the internet infrastructure in terms of redesigning and redeploying.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Very valuable input, @Bogdan.Warinschi . Thank you.

Could we replace the sentence "This has to be avoided." with something more nuanced and would this address your concerns? If yes, could you make a proposal?

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Yes, sounds good -- let me try to come up with a replacement sometime later today.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

@christian.folini Below is a (perhaps too wordy?) draft of a replacement for the summary. I've tried to incorporate the suggestsions by @Carsten.Schuermann and @Olivier.Pereira . I'm a bit affraid that the trade-offs may be obscured by the lengthy descriptions. Perhaps we should iterate a bit if, for example, we can reach agreement that it is reasonable to proceed without worrying too much about quantum computers? In any case, happy to discuss re-wording, further clarifications, etc.

Voting secrecy is a basic right, guaranteed by law. If a Public Board raises doubts about this guarantee, then this undermines not only trust in the Public Board but also in the internet voting channel and voting as a whole. The use of a Public Board raises some concerns regarding vote privacy. Depending on the cryptographic techniques used, advances in cryptanalysis, or due to bugs, the information on the board may reveal how a voter voted.

While one can question the benefit of this information for these actors in the possibly distant future and the benefits look rather limited in that light (both for embarrassment and coercion of the voters) it is important to understand the risks and trade-offs associated to using a Pubic Board, and different designs thereof.

Some Public Board designs publish encryption of votes. Here, secrecy of votes relies on the same set of assumptions which guarantee secret communication over the internet. Yet, these assumptions may be invalidated, and votes may be decrypted if quantum computers become a reality. If the board contains voter identifying information (e.g. required for auditing purposes) then the link between voters and their vote may be revealed. These designs may potentially be strengthened by using encryption schemes which are secure even against quantum computers -- such schemes are under development by the cryptographic community.

Other designs may hide the link between voters and their encrypted ballots so even if ballots get decrypted individual choices stay secret.

Finally, other designs aim to achieve "everlasting privacy": instead of publishing encryptions of votes, such schemes publish a so called "perfectly hiding commitment" -- essentially a hash of the encrypted vote. By themselves, such Public Boards reveal absolutely no information about the votes. Of course, the overall guarantees for vote privacy still relies on the security of the rest of the building blocks.

Edited by Bogdan Warinschi 5 months ago

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Hi @Bogdan.Warinschi , Thanks for this pass! A few reactions:

- "the information on the board may reveal how a voter voted e.g. due to significant advances in quantum computation." Would it make sense to clarify at this stage that this really depends on what is put on the board -- computationally or perfectly hiding commitments -- and not an inherent feature of any bulletin board (that is, in theory, if we have perfect randomness and perfectly hiding commitments (and...), then we know that the board will not leak)? Here is an attempt of rephrasing: "depending on

cryptographic design choices, or as a consequence of bugs, the information on the board may reveal how a voter voted"?

- Maybe I am missing something, but I am not sure that I understand why, in this discussion (last 2 sentences), we should care about the properties of the communication channels used to submit the vote. What you are writing is of course perfectly correct, but it is a concern that is independent of the existence of a public bulletin board. So, I wonder if this may just distract from the question of the privacy concerns raised by such a board.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

@Olivier.Pereira thanks!

- yes, my statement was not sufficiently nuanced, but what you suggest is what i had in mind.
- this designed description was a bit waffled -- I've attempted a rewrite

Edited by Bogdan Warinschi 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for this new summary @Bogdan.Warinschi, including the update after Olivier's input. I think it covers all the essential arguments and does a better job at being systematic than my initial draft.

Also nothing wrong with being longer. This is obviously a very important topic and I think it is a good idea to present different design variants side by side.

Edited by Christian Folini 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

@Bogdan.Warinschi just FYI, your link to eprint 2017/1235 is broken because you accidentally included an extra close-bracket as part of the link.

Bryan Ford @Bryan.Ford · 5 months ago Developer

@Olivier.Pereira to respond to your second question above, I think it's important to point out that even a "baseline" E-voting design today without a public bulletin board necessarily exposes in-transit encrypted votes to potential recording and future decryption by an adversary, to provide the reader a better understanding of the constellation of "eventual-decryption" risks and make it clear that a public bulletin board is not the only element in which such risks can be found. Otherwise, an uninformed reader might readily jump to the conclusion, "ok, let's just kill the bulletin board idea and we're totally safe against quantum computers, right?"

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

@Bryan.Ford: Yes, there should be a proper safeguard against that! I believe that @Bogdan.Warinschi 's sentence "Of course, the overall guarantees for vote privacy still relies on the security of the rest of the building blocks." does the right job here.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Apologies for this comment late in the game. Instead of encrypting the vote, one could also think about hiding the identity of the voter, for example, using a ring signature scheme. The idea is to group all voters into groups of a 1000 or so. Using a suitable ring signature scheme, one would allow each voter to sign the vote with a signature (that is then stored on the board), however, this signature can *not* be traced back to the individual voter, but only to the group of voters. See EvoteID'17

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Carsten. @Bogdan.Warinschi has issued a new draft summary above.

I think your idea is included in the following very broad sentence in the new summary. Correct?

> Other designs may hide the link between voters and their encrypted ballots so even if ballots get decrypted individual choices stay secret.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Hi, and thanks Christian for this summary! Two extra points which, I suspect, can be important:

1. I believe there are good reasons for inviting people to publish/reveal their "ballot fingerprint" as it is displayed on the bulletin board, linking it to their name. Here are two:

    1. A technical one is that it can contribute to filling an important gap in verifiability. Having a public bulletin board with no link between names and (encrypted/committed/...) votes leaves open the possibility that malicious/corrupted authorities add extra votes on the board (i.e., perform ballot stuffing). It would make sense to keep possibilities of investigating this, and it seems then important to encourage people to reveal their ballot fingerprint for auditing purpose. (One such process that we used was to give to appointed auditors a list matching voter names and ballot fingerprints, and to allow these auditors to contact a random sample of the population in order to ask them to confirm their fingerprint as it appears on the board, or to confirm that they did not vote by Internet.)
    2. A related point, but maybe more social, is that it seems important to communicate to the people that what appears on the board does not leak anything about their vote. As such, it seems contradictory to suggest that people should not publish their vote fingerprint: either there is a risk, or there is no risk. And, from a verifiability point of view, it makes sense to actually encourage people to publish their fingerprint.
2. Protocols with everlasting privacy do not solve everything: everlasting privacy is still conditioned on the use of uniform randomness. And we saw so many bugs with PRGs (including in real-world election -- see the Norwegian case) that we cannot rule out that a public bulletin board could amplify the damages caused by a bad PRG used by a voter to encrypt/commit to their vote. This is orthogonal to long-term security, breaking DDH, or post-quantum crypto: whatever solution is used, if a voter uses a bad source of randomness, he is leaking his vote. And the problem just looks worse if the poorly protected vote appears on a public board rather than remains in the hands of authorities who we can hope to be honest.

(One trick that Helios uses (and possibly other systems) in order to mitigate this last issue is that the web server sends to the voting client a sequence of random bits, together with the web page. The client then" mixes" its own randomness with the server randomness in order to produce the randomness that it uses to protect the vote. This means that: (i) if the client local randomness is good, we are safe (ii) if the client randomness is bad, a malicious server might learn the vote but, if the server provided good randomness, then the encrypted vote displayed on the bulletin board still offers protection against third parties.)

Edited by Olivier Pereira 5 months ago

Collapse replies

Christian Folini @christian.folini · 5 months ago — Maintainer

Thank you @Olivier.Pereira. Are these concerns covered in Bogdan's new summary from your perspective?

If not, then please shout and we'll give it another pass.

Olivier Pereira @Olivier.Pereira · 5 months ago — Developer

That looks all good to me, thanks @Bogdan.Warinschi, thanks @christian.folini!

Christian Folini @christian.folini · 5 months ago — Maintainer

Marvellous. Thank you.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago — Developer

@christian.folini: will there be a phase were we can refine these conclusions?

Christian Folini @christian.folini · 5 months ago — Maintainer

I definitely see the need for this. But this is overridden by the schedule and the many, many open questions that we need to answer until July 10. We have to be done by then and if we want to refine conclusions, we need time for everybody to think about a new wording and confirm it. This takes too much time.

But here is what we are going to do: We will wrap up the whole block 4 into a single large response / summary, likely under 4A. This will include all the different summaries we are developing separately. This might look a bit untidy, but I think it's a good pragmatic approach to allow you to see it all in context.

Vanessa Teague @Vanessa.Teague · 5 months ago — Developer

I agree with @Olivier.Pereira, especially 1.1 and 1.2.

Christian Folini @christian.folini · 5 months ago Maintainer

Following concerns with my draft summary, @Bogdan.Warinschi has written a new summary for this question. He refined this after input and we have now adopted it as *the* summary for this question above. Think this new summary has the support of everybody in this discussion here.

If you do not agree with the new summary, then please be quick. I plan to close this discussion tonight.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I suggested above a hopefully more technically accurate but still clear phrasing for the "essentially a hash" issue, which I wrote before I'd seen Bogdan's new summary, but I think my suggested paragraph still works as a drop-in replacement for that paragraph in Bogdan's summary.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Yes, I think it still does and it replaces my poor analogy with a better description. So I have adopted that change as well. Thank you.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Yeap that's much better!

Christian Folini @christian.folini · 5 months ago Maintainer

OK, here we go again. The previous update stirred the discussion even more, so I have edited the summary twice tonight. Please check the latest version above.

Tomorrow will be my next attempt to close this discussion. :)

(And please do not get the feeling I am getting annoyed. I am learning a big deal every day here. So please go on until you are really satisfied with the summary.)

Edited by Christian Folini 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Latest version looks fine to me!

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bryan.

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Closed Opened 6 months ago by Christian Folini Reopen issue

Discussion 4F - Public Board to determine voter turnout early (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

According to both propositions, the full contents of the PB would be readable by the public at any given time. Thus, the preliminary turnout would be known before the voting period ends. This could be avoided by limiting access to the parameters and to the data related to the individual voters before the voting period ends. However, we wonder whether knowing the preliminary turnout should be considered an interesting feature rather than a problem. What do the social scientists think?

Drop or upload designs to attach

Linked issues 0

Christian Folini @christian.folini changed due date to April 06, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

Is the idea here to duplicate some of the functionality of exit polls?

Reto Koenig @Reto.Koenig · 6 months ago Developer

In our opinion the question is ok, but the proposed restrictions on the PB are not allowed. If the PB is not oblivious to the requester (hence knows who requested what data), it is given too much power to tamper with the data (the append-only property would be at stake).

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

I agree, either you have a fully accessible public bulletin board or not.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

During an election in DK, in every polling station and at every hour, voting percentages are computed, compared to the previous election, and announced. Therefore, 4F is more a question for political scientists and not so much for the public bulletin board. The question shows, however, how important it is that the public BB is properly encrypted (not as in Norway 2013): If partial results are leaked (what would have happened in Norway were their data public), the election could no longer be considered fair, violating international standards.

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

@Carsten.Schuermann , in Denmark, does this apply to election day only or also to an advanced voting period (if there is one)? That is the crucial question for the Swiss case where we are dealing with 3-4 referendum votes per year with an advanced voting period

of three weeks.

Edited by Uwe Serdult 5 months ago

Bogdan Warinschi @Bogdan.Warinschi · 6 months ago Developer

Preliminary turnout data is not a secret. In fact, for standard elections, turnout information is regularly broadcast as a matter of public interest so I see no issue with the PB leaking it.

In this context, as @Carsten.Schuermann points out, "no partial results leak" is a crucial property. The property is often mentioned, but I don't think it had been formalized on its own. I'm confident however that it follows from any reasonable definition of vote privacy.

Christian Folini @christian.folini · 6 months ago Maintainer

I just learned that the city of St. Gall publishes the number of votes arriving via mail daily. Link

Nice article btw way.

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

The City of Zurich does it as well (with a daily updated ticker). However, one is not able to know for which of the respective referendum topics (usually several are bundled in Switzerland, from all levels, national, cantonal, municipal). Upon request by the press most municipalities give out such data on a regular basis.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I'm not a social scientist, but I agree with the points already made that while preliminary election *results* are sensitive and should not be released, I don't see any reason that preliminary *turnout* data should be considered sensitive, and there may be many reasonable uses for such turnout data.

Fabrizio Gilardi @Fabrizio.Gilardi · 6 months ago Developer

I'm not sure that the full contents of the PB can tell us anything about turnout. Only a (small?) subset of voters are likely to use the PB, and those voters will not be representative of all voters. That said, someone might try to interpret PB records as reliable indicators of turnout, even if they are not.

Yet another reason why the PB is likely to create confusion among voters.

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

My understanding was always that each single electronic vote is automatically written onto the public bulletin board. Otherwise public bulletin boards do not make much sense to me. However, I might be wrong there. Tech experts please correct me if I am wrong.

Bryan Ford @Bryan.Ford · 6 months ago Developer

Your concern points to a potentially-important design question: do *all* vote ciphertexts (or commitments to them) get posted on the PB, or just the ciphertexts of voters who actually use the verification function?

I personally think the former approach would be preferable, since it would offer maximum transparency for verification purposes. Otherwise voters won't have any hope of being able to verify total vote tallies from the information on the PB for example, and it would be really nice (I think almost essential) if they could.

If all vote ciphertexts do go on the PB, then it seems like the data on the PB probably does represent voter turnout accurately.

I can see potential arguments for making only "verified" vote ciphertexts go on the PB. For example, if having the ciphertext posted and verified is seen as creating a larger privacy risk (e.g., against long-term vote decryption) than not posting a vote ciphertext. However, I think the correct solution to this kind of privacy concern is - as @Vanessa.Teague and others pointed out in 4E - to post information-theoretically-hiding commitments and ZK proofs on the PB instead of vote ciphertexts directly. With that kind of PB design, I don't see any obvious reason why we wouldn't want at least commitments to *all* cast votes to appear on the PB for maximum transparency and user-verifiability.

Florian Egloff @Florian.Egloff · 6 months ago Developer

The question raised here would have to be clarified.

Also: @christian.folini raised the fact of a City listing the number of votes arriving via mail daily. Perhaps the existing diversity should be seen as an inspiration to allow flexibility of whether a canton/municipality wants to have a public notice board or not?

For me, the most crucial point is that election results are not disclosed before the end of the election/referendum: if that property cannot be upheld with a public notice board, then that would rule it out.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for contributing to this discussion. It's time to wrap it up, so here is my attempt at a summary:

According to both propositions, the full contents of the PB would be readable by the public at any given time. Thus, the preliminary turnout would be known before the voting period ends. This could be avoided by limiting access to the parameters and to the data related to the individual voters before the voting period ends. However, we wonder whether knowing the preliminary turnout should be considered an interesting feature rather than a problem. What do the social scientists think?

It is very important, that any form of Public Board does not leak preliminary results of a vote or election. Data, that allows to deduce voter turnout is less of a concern, though. Also because this exists for voting by mail in some Swiss voting circles already.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

[EDIT]: Removed 2nd paragraph ("If a design would be chosen, where only the vote ciphertexts (or commitments to them) of voters who actually use the verification function are published via the Public Board, then deducing the voter turnout might be misleading.") as it is not the core problem this question deals with and it would be discussed if a public board would be adopted quite naturally. See comment by @Bogdan.Warinschi below.

Edited by Christian Folini 5 months ago

Collapse replies

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I would reword "It is very important, that any form of Public Board ...." and say why it is important. The reasons is, that Switzerland is bound by international commitments (https://www.osce.org/odihr/elections/14304) (6) to hold a fair electoral processes, where fairness refers to the fact that no preliminary results are leaked.

Christian Folini @christian.folini · 5 months ago Maintainer

@Carsten.Schuermann : I thought about this for several days. I think the wording "It is very important" is good enough. (a) because the Swiss Federal Court made it very clear publishing early is not OK and several Swiss municipalities had to drop their habit of publishing 1-2h early and (b) if there is clear legal practice in Switzerland, using international treaties often weakens an argument in the political context. So it's not *very important* because the OSCE says so, but because the Swiss Federal Court says so.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

I would add: "Just like there is diversity in publishing turnout before the close of a referendum/election, one should explore the possibility of giving communities choice in whether they would like turnout to be published earlier or not."

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Florian: This is about active publication is, not it? You are not stating anything with regards to 3rd parties *deducing* turnout from the published data, are you? I think this matters, because if you want to prevent 3rd parties from deducing that, you need to chose a design that prevents that and we ought to make that transparent, I think.

Florian Egloff @Florian.Egloff · 5 months ago Developer

No that is not what I meant. Edited for clarity: "Just like there is diversity between municipalities in publishing turnout before the close of a referendum/election, one should explore the possibility of giving election/referendum organisers (e.g. cantons) the choice in whether they would like turnout of the internet voting channel to be published earlier or not."

Is that clearer? I.e. if St. Gallen wants to publish as votes come in, they should be able to, but if Appenzell decides they do not want that functionality, they should not be forced to do so. This should remain a democratic choice and not be a "forced" technologically determined outcome.

Christian Folini @christian.folini · 5 months ago Maintainer

OK. I see that. However, they way I read the discussion, this is based in the design of the PB. Once the information is put on the bulletin board, anybody can extract the necessary information to calculate the turnout. There is no additional publication necessary. So the way I understand it, giving municipalities that choice could essentially mean to make give them choice over using the PB or not and I do not think that is viable.

What could be done is of course give the municipalities the choice to extract the information and publish it in a readable form themselves and quasi-officially and municipalities could then do that or not.

Did I describe this correctly, or am I missing something here?

Florian Egloff @Florian.Egloff · 5 months ago Developer

I think we disagree with the characterisation of "viable".

Is there a reason to force every canton to use a Public Bulletin board? In my reading of the discussions, there are good reasons to do so, but it is unclear that it is a necessity(i.e. in the strong form of "only if there is a public bulletin board, voters are justified in having confidence in an internet-based voting system). Thus, as a political scientist, I am inclined to think that the existing diversity in publishing turnouts could be part of insitutional inertia, but could also represent different (local) ideas of how democracy ought to work. In that sense, I would encourage including localities in the design choices of the system operating in their jurisdiction, and only mandate from the Federal level what is a "strict" necessity.

Christian Folini @christian.folini · 5 months ago Maintainer

This is a new idea. I think nobody in this discussion thought of making it a choice for the canton. The line of thought for the question (and the other responses in this block) have been, a PB becomes mandatory and everybody gets one. Or a PB is an exotic non-Swiss beast that no system provider would build unless it's mandatory.

I'm adding this in a separate comment below and wait for feedback if we want to take it into the summary or no.

Can we agree that it's not going to be part of the summary if none of the other experts thinks it's a useful choice?

Florian Egloff @Florian.Egloff · 5 months ago Developer

if other experts think it is a "necessity", it should not be part of the summary. If, however, other experts think that it is reasonable (and secure enough) to let the political units decide for themselves whether or not to have a public board, I would leave it in.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you. Let's see if the idea gets any traction below.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I like what @Florian.Egloff suggests -- letting cantons try different things with the PB. It's a typically Swiss approach with clear potential benefits in terms of learning. The key is to agree that the PB is not a key component of security or trust, which I think it is not. It's one tool that, if designed and used correctly, might be useful. There is no clear best practice at the moment, so letting the cantons experiment may be a good idea.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

I'm not sure what the second sentence wants to say/warn about. It seems to hint at a design where a voter cast a vote but it's vote is only placed on the bulletin board when the voters runs some verification protocol.

Such a design is not common (as far as I know) and I don't see the need to be so specific. For example, one can imagine a design where votes are published in blocks to hide, say, the distribution of time between votes cast. Then a similar caveat would apply.

(NB: i've moved this comment down from the replies to the message from @christian.folini setting out the conclusion. It may be useful to better see the chronological relation between the issues raised, and address them in order)

Christian Folini @christian.folini · 5 months ago Maintainer

@Bogdan.Warinschi : Are you referring to the 2nd sentence or the 2nd paragraph? I think it is the latter, but I want to be really sure before I respond.

Collapse replies

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

@christian.folini You're right: I mean the 2nd paragraph

Edited by Bogdan Warinschi 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Good. The 2nd paragraph picks up a concern that @Fabrizio.Gilardi (upvoted by @Florian.Egloff) raised above. @Bryan.Ford has addressed this and opts to publish all information.

I was not aware that such a design was uncommon. So this caveat might indeed be too specific. I see two courses of action:

- We drop the 2nd paragraph as it addresses a niche problem and this would be discussed anyways (if a PB would ever be introduced)
- We add a sentence to this paragraph. Proposal: "It is therefore preferred to publish information for all voters."

I think dropping would be preferable, because it is really a side track and should not get too much weight.

However, I'd like to hear more opinions before I make such an edit.

Christian Folini @christian.folini · 5 months ago Maintainer

Dear @Fabrizio.Gilardi, @Oscar.Nierstrasz, @Bryan.Ford, @David.Basin, @Reto.Koenig, @Oscar.Nierstrasz, @Bogdan.Warinschi and @Vanessa.Teague,

@Florian.Egloff has brought up the idea that a PB should be an optional component that a canton can choose - or not. Thus not a mandatory requirement in the VEleS, but an optional design variant that a system provider can offer at his own discretion. Please take a minute to read through his arguments.

Do you agree that this is a useful choice, or would you rather make it a full requirement - or drop the idea completely?

Please respond to this question, as it's really important for the wording of the summary above.

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

I think there is not much use of having universal verification without a public bulletin board. The public bulletin board is the very component opening up the black box of internet voting to whoever in the know and capable of building a verification tool independently of authorities. The public bulletin board lifts transparency to a new level allowing not only members of the municipal election committee to observe and control the vote result within their own municipality. It allows to actually verify all electronically cast votes, also outside of the municipal borders.

Florian Egloff @Florian.Egloff · 5 months ago Developer

Sorry to be a pain. Probably the discussion would properly belong into #20 (closed), but the intuition was triggered by the turnout publication.

I also see the value of a public bulletin board; I am just not sure that I read other people's answers to indicate that it is a "necessity" in the sense outlined above, thereby raising the question whether it has to be mandated by the federal government. Perhaps the federal government could mandate that any system in place has to offer the option of a public board at no additional cost instead of mandating that it has to be used by the cantons? The idea is that this would ensure that the development cost has to be borne anyway and a race-to-the-bottom in terms of security is avoided.

Edited by Florian Egloff 5 months ago

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

I see a problem with cantonal decisions on PBs. Having a canton telling its voters that they have PB because it is good for security and another canton saying they don't have it because of some reasons would be difficult to understand and detrimental to trust.

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

I strongly agree with this point

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I see the problem, but it seems rather hypothetical to me. Is there any evidence that some Cantons would object strongly to the use of a PB? I sympathize with the idea of involving them in the discussion of the requirements and design, but am also against making the PBs optional, as this would give the public a mixed message.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

Is there any evidence on what the cantons think about the PB? The discussion here shows some divergences regarding the usefulness of the PB, so I wouldn't be surprised if different cantons had different views.

I see the problem with mixed messages. But the specific implementation of e-voting is up to the cantons, so it's not like everything will be identical in all cantons. The PB might be just one of those things that are done differently depending on the canton. It doesn't necessarily have to stand out as confusing in terms of trust and security.

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

In the end, the cantons are the ones who pay for the system. I guess they will thus think in terms of cost/benefits. They will only invest in a PB if they have a convincing argument. If the argument is convincing then we're back to the mixed messages sent out by the cantons not using it.

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

I disagree with this - maybe I was too quick to agree above. Not all arguments are "convincing" to everyone. I can imagine cantons choosing different types of implementations.

I see our role to ensure that the security trade-offs are transparent and that making choice available should not prevent a more secure/transparent solution from being realized (i.e. no race-to-the-bottom should not be triggered). Thus, that is why I think it may be sensible to separate recommendations with regard to what operators have to offer from what cantons can choose vs. have to implement.

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

I fully agree that not having a public bulletin board would be detrimental to trust. Trust is not only a desirable feature based on security and trust in the ones who distrust (the ones building tools to verify the information in the public bulletin board), it also translates into higher user rates. The last thing you want to end up with is having to deal with lower internet voting user rates in cantons without public bulletin boards versus consistently higher ones in cantons with them.

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

Enabling the detection of manipulated votes is clearly a strict necessity. (What should the Confederation ask for if not exactly that?) If a PB seems to bring significant added value for the ability to detect manipulated votes (and thereby for the credibility of the result when no manipulation is observed), then corresponding Federal requirements would be the natural consequence. It would not be some fashion-choice left to the cantons. There might be other reasons to have a PB, but a PB without any meaningful security argument is not what we're looking for. (Leaving the cantons the option of introducing a PB that is declared as not serving security is somewhat a non-scenario.)

Another thought. Assuming that technically and financially we have the chance to introduce a PB and thereby make it harder for "the system to cheat", thanks to components outside of the system. Couldn't the public perceive it as suspicious (bad for trust) if such a PB were not introduced/required? Couldn't the result be put into question just from the mere fact that a PB could have been introduced to challenge the system, but an active decision was taken not to? Which argument would be convincing at explaining why a PB was not introduced?

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

The problem is that the added value of a PB for security hinges on the public's behavior. Security does not increase automatically just because a PB is out there.

Unless the PB is politicized by parties, NGOs, or interest groups (it may or may not be -- we don't know at this point), the vast majority of voters won't notice or care if a PB was introduced or not.

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

@Fabrizio.Gilardi, sure the added security depends on the voters' behaviour. But isn't it reasonable to suspect that even a small subset of voters would probably care at some point (use the PB if it's there and maybe even miss it if it's not)? Even a small group could detect large-scale manipulations. Of course the administrations would have every interest to continuously promote the use of PB over time, given that it would add to the ability to detect manipulated votes. I like to believe that most voters are reasonable people. Accepting a priori that despite voter education not even a small group would care about using a PB, is hard for me to do (but I might be wrong here, not a political scientist).

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

Could you both @Fabrizio.Gilardi and @oliver.spycher please be more precise about what you mean when you say that it depends on the behaviour of the voter. I think there is a misunderstanding here.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I'm sure some voters will use the PB with the best intentions. It's possible others will use it also with the best intentions but making mistakes, leading to rumors on social media that there's a problem with the election. That wouldn't be good.

Collapse replies

Uwe Serdult @Uwe.Serdult · 5 months ago Developer

How exactly will the voters use the public bulletin board?

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

Doesn't this lie in the very nature of verifiability? -The more instruments that are available to enable verifiability, if done right, the higher the chances to detect manipulations. This goes somewhat hand in hand with increasing the risk of false claims and confusion. And that holds true for the existing return code mechanism too that was used over years. Should verifiability go away in order to make false claims go away? Regarding rumours in social media: Couldn't the absence of verifiability lead to just-as-nasty rumours about security issues just because of the lacking verifiability? -I would prefer the rumours in presence of verifiability, not in absence, because only with verifiability you can detect manipulated votes.

David Basin @David.Basin · 5 months ago Developer

Whether or not there can be false claims depends on what is posted on the bulletin board and whether disputes about claims can be resolved. One can never stop false statements, but ideally one can resolve them.

Dispute resolution is a property of the protocol involved: A dispute can be resolved if (in case of disagreements) any third party can unambiguously determine who is right. Let's take some simple examples. * If all the votes end up in plain text on the bulletin board (after mixing and stripping of identifying information) then no disputes can arise as to the tally as all can compute the tally themselves.

- If NIZKP are appropriately used, nobody can dispute that the plain text votes corresponds to the cipher text votes.
- Via appropriate uses of signatures and encryption (e.g., See the Mixnet voting protocol proposed in https://arxiv.org/pdf/2005.03749.pdf) it is possible to reconcile individual verifiability and dispute resolution.

In sum: whether a PB can lead to rumors and misinformation depends on the voting protocol. Perhaps a solution to this problem is then to insist that if PBs are used, then the protocol involved must allow disputes to be resolved. (And this is checked when checking the protocol design.)

Oliver Spycher @oliver.spycher · 5 months ago Maintainer

@Uwe.Serdult , @Fabrizio.Gilardi , let me clarify. I agree that a PB can only add to security if a sufficient number of voters use it correctly (e.g. by scanning a QR-code on their laptop screen and reading on their mobile what they are about to vote for as well as a confirmation later that they have voted that way. Btw is this more complicated than today's return-code scheme?). I also agree that mistakes by voters and false allegations are serious issues to consider and I believe appropriate means of communication, procedures for conflict-resolution and verification processes that focus on usability would need to be elaborated. My point is that even if there are rumors and false allegations (which actually I don't see depending on the presence of a PB), the PB would have been announced to serve (and would truly serve) the detection of manipulated votes and therefore something presumably important to many voters. This is why I agree with those arguing that introducing a PB should not be an option the individual cantons can decide for or against.

Edited by Oliver Spycher 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

Edit: Moved the following comment to #20 (comment 968)

I like that @David.Basin raises dispute resolution as the key.

I would expand on @Fabrizio.Gilardi 's comments above: an attacker attacking the legitimacy of elections only needs to attack the appearance of trustworthiness of the process or outcome, not the actual trustworthiness. If the elections can be made to appear to be "rigged" then the attacker has a chance to disrupt the fundamental value in democracy, namely, that the losers accept the outcome. Internet voting has the potential to introduce such a means of sowing doubt. This has nothing to do with the objective "security" of the system, but rather, with the social familiarity & trustworthiness of the system by the population: if the population does not have trust in the dispute resolution before the election takes place, it will be hard to convince a population of their "trustworthiness" in the actual event.

The relevant question thus is not "do you trust internet voting enough to use it?" but rather, "given a scenario of widespread allegations of abuse and cheating, do you trust the process designed to prove to you the trustworthiness of the process and the result?" Thereby, I assess, it will be very hard to convince an (adversary stoked) distrustful population that one should trust the authorities' claims that the technical systems worked properly, that the programs claimed to be running were running, and that the outcome is correct. This is not a remote possibility, but rather the baseline threat any change of democratic voting has to deal with.

Thus, I suggest, we should discuss the issue of dispute resolution, and the process of how voters form trust in it, here.

Edited by Florian Egloff 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you everybody for the additional input. This question here, originally seen as a side track, proved very fruitful for the discussion of PBs far beyond the voter turnout question.

Here is the plan how to close the voter turnout question at hand here:

- We are taking the minority position to make a PB a choice for the Cantons into the overarching discussion / summary of 4A. So that is not being entered in the summary above, but in the one in 4A.
- The importance of the dispute resolution has become so apparent, it should not be discussed / covered in the summary above, but also in the overarching discussion in 4A.

I have dropped the 2nd paragraph of the original summary above for the narrow 4F question. If I do not hear any additional opinions, I will close 4F tonight.

Christian Folini @christian.folini mentioned in issue #20 (closed) 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Thanks Christian. Fully agreed with the sentiment that social trust and dispute resolution are extremely important; I've followed up in 4A.

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 4G - Public Board and support for voters (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

Following the logic of querying a PB with independent software of own choice, would it be correct for the cantons or the provider not to offer support in case of problems during the verification process?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 07, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

I think I do not understand the question. Not offering support in case of any type of problems seems quite strange to me.

Christian Folini @christian.folini · 6 months ago Maintainer

It is indeed quite strange but there is some logic to it. If the PB is meant to be an independent instrument that allows to supervise the correct operation of the vote, then the cantons and the provider are the wrong entities to provide support for the voters (especially in the situation where a voter detects a malfunction and wants to report it ...).

On the other hand, it is very hard to think of a 2nd or 3rd point of contact for the voters in case of issues with using the PB (and only for this particular situation).

So if you think this through, then providing support is really tricky.

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

You can't have an independent instrument without an independent body to run and support it. You can't trust the instrument if you don't trust the body. This sounds like a recursive can of worms to me.

Bryan Ford @Bryan.Ford · 6 months ago Developer

I feel the question is a bit vague, specifically the phrase "not to offer support": not to offer support for what exactly?

My reading of the question's intent is that if the E-voting system includes a PB providing the ability for voters to verify their votes independently using independently-developed software *not* originating from the E-voting provider or the cantons, is it reasonable for the cantons and E-voting provider not to offer "customer support services" for voters' use of that independent third-party software?

Assuming this is the correct reading of the question, then I do think this is a reasonable policy, and in some sense necessary and inevitable. No software or service provider can reasonably expected to promise to provide "customer support" for other independent software they did not create or vet and have no control over.

If the verification software the voter is using is provided by some third-party vendor, then any customer support services provided for that verification software should naturally come from that third-party vendor. If the verification software is open-source software developed by a random person on the Internet and posted to GitHub, then anyone using it must do so with the usual "use at your

own risk" and "no warranty or promise of customer support" caveat, just as with all other software one might download and install from a random developer on GitHub.

However, I do think it might be reasonably expected for the cantons and/or E-voting provider to provide something like the following more restricted forms of "support" for verification tools:

1. The E-voting provider should probably offer a (fully open-source of course) "reference implementation" of a verification tool, which they might reasonably provide support for, given that it comes from them.

2. The E-voting provider and/or cantons might attempt to create and maintain a list of third-party verification tools (apps or whatnot) that have some commitment of support by their respective third-party vendors, and have received some threshold of independent expert peer review. This list wouldn't be a promise to support the use of those tools directly or necessarily even an endorsement, but rather just an aid to voters looking to choose and make use of independent third-party verification tools.

As I've argued before in other discussions and will continue to argue, diversity is good - but the support structure and associated costs also need to reflect that diversity appropriately.

David Basin @David.Basin · 6 months ago Developer

Maybe the author of this question can clarify its intended meaning? If the meaning corresponds to Bryan's reading, then I support his answer.

Christian Folini @christian.folini · 6 months ago Maintainer

I think Bryan's response follows the explanation I tried to give above. He is thus right on target with his interpretation of the question.

Unfortunately, Oscar's statement about the *recursive can of worms* has some truth to it. That's why we included this question. Sorry if the wording was not clear enough.

Carsten Schuermann @Carsten.Schuermann · 6 months ago Developer

One comment I would like to add to this discussion adding to @Bryan.Ford 's answer is that the protocol and API must be precisely defined and public and stable for an open source verifier to work. However, there is another attack vector hidden here, which I think is worth mentioning: One way to render all verifiers unusable is to change the API last minute, by adding or dropping a byte here or there. The cantons might not be responsible for the verifier app, but they are responsible for making sure for following the rules of update and maintenance. If not, trust can be easily destroyed.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for this idea. It did not occur to me so far. I'd rather not include it in the summary of this thread, but we must not omit it either. It is probably best to include it in 4B. I have placed a reference in 4B pointing to your comment here.

Florian Egloff @Florian.Egloff · 6 months ago Developer

I think at the very least, the cantons would have to provide "support" for others to use the verifier that they are using themselves.

Christian Folini @christian.folini mentioned in issue #21 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for contributing to this discussion. It's time to wrap it up, so here is my attempt at a summary:

Following the logic of querying a PB with independent software of own choice, would it be correct for the cantons or the provider not to offer support in case of problems during the verification process?

It is important to distinguish different support situations:

1. Support with using a verification software that is officially endorsed by the canton.
2. Support with using a 3rd party verification software.
3. Support for a voter when the software fails to verify the vote and a dispute arises.

(1) A canton needs to provide support for voters using officially endorsed software.

(2) In case of 3rd party software the canton should maintain a list and in case of problems the canton should be able to direct the voters to the provider supporting the software.

(3) If the role of the Public Board is to verify the correct operation of the vote by the system provider and the canton as well. Then the canton is not the best contact in this situation and there has to be an independent support provider. However it is difficult to design a solution that addresses this case adequately: If the bad reports are escalated before they are thoroughly confirmed, then the trust could be undermined without good cause. And if the support provider discourages the voter from pursuing a report, then a malfunction or fraud could go undetected. It is therefore important to design and implement proper dispute resolution for this and other situations where a dispute may arise.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is none or no negative feedback I will sooner or later assume consensus and close this discussion.

[EDIT] Updated last sentence after feedback by Carsten Schuermann.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I believe that (3) is really important. But I also know that failure to verify the PB is not the only place where disputes may arise. (E.g. return codes don't match, error messages while casting a vote, etc.) Therefore, I would suggest to design a general dispute resolution mechanism (which is possibly already described elsewhere) and refer to it here.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

We're entering federal territory here. Dispute resolution is the authority of the cantons and it's unlikely that FedCh could implement a general rule that works everywhere.

Do you think there is a need to update the summary, or can we leave it like this?

Christian Folini @christian.folini · 5 months ago Maintainer

@Carsten.Schuermann : I have adjusted the final sentence of the summary based on your input. I think it is more readable now and integrates *this* conflict with the other situations where dispute resolution is necessary. If I do not hear from you, I will close this discussion tonight or so.

Christian Folini @christian.folini removed Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 4H - Public Board vs. Independent Control Component (see Block 2) and adapted parameter generation (see Block 3) (Block 4 - Public Board)

Reference to originating discussion block

Block 4 - Public Board

Question

Recalling Block 2, would you recommend Bob to introduce a PB or rather to procure new control-component software or adapt the protocol to achieve parameter generation that is more trustworthy, assuming he could only pick one?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to April 08, 2021 6 months ago

Christian Folini @christian.folini added Block-4 Cryptography labels 6 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 6 months ago Developer

This sounds like a Software Engineering question to me. What do you want to optimize? What non-functional requirement are you prioritizing? If it is a question of earning voter trust (and not just a technical security issue), then you may need user studies to answer the question. But first you must clarify exactly what *measurable* goal is to be achieved here. It is not so clear to me.

Reto Koenig @Reto.Koenig · 6 months ago Developer

As already mentioned in block 3A, the public parameters should be fixed and always accessible via the publicly available specification (signed by an authority such as BK). So, we do not see any advantage in publishing public parameters again on a PB.

Christian Folini @christian.folini · 6 months ago Maintainer

The summary of the 2nd discussion block on diversity is resulting in a very extensive list of additional requirements for the control components. Block 4 - this block here - proposes a public bulletin board that can be perceived as a very different approach or even as a base for a different trust model.

@Oscar.Nierstrasz is quite right in asking for measurable goals. However, we can not provide them (as of this writing). The plan is to get answers to the (political) question what is more important, which one is more beneficial, what do you recommend? There are no straight answers here obviously and user studies might help indeed. But we are asking for your opinion and your preference without all this useful information.

If you have a preference at all. It is perfectly OK to say that you think both should be required. However, through this discussion block, various experts reported a lot of different downsides and problematic aspects of a PB, so it has become quite difficult to understand just what the big response really is: (1) A PB is very difficult to do and the privacy implications are insurmountable, so you better don't even think about it and concentrate on truly independent control components instead. Or (2) a PB is hard to get right and you need to be really careful, but all things considered it is a useful and valuable element of an internet voting system.

The various downsides expressed here contrast with the more positive response the PB received in multiple responses throughout the questionnaire. I'm thus inviting @Vanessa.Teague, @Olivier.Pereira, @David.Basin, @Srdjan.Capkun, @Rolf.Haenni, @Philipp.Locher, @Reto.Koenig, @Uwe.Serdult, @Florian.Egloff, @Bryan.Ford, @Carsten.Schuermann and @Ulrich.Ultes-Nitsche to return to this discussion block and ponder over this particular question to provide us with some guidance if a PB should be pursued at all.

(I am not stating that everything was blue sky with the PB in the questionnaire, but the positive aspects got more weight than they got here.)

Bryan Ford @Bryan.Ford · 6 months ago Developer

I think that (a) investing in diversity through independent control component software, and (b) investing in greater transparency by introducing a public bulletin board, are both critical enhancements that I see as co-equal with each other in importance: both should absolutely be done. I see more (c) trustworthy parameter generation processes as useful but clearly lower-priority than (a) and (b).

In particular, I feel that investing in more implementation diversity (a) as absolutely essential to providing a big improvement in real and measurable security, in part because this is the only approach I'm aware of that can quantifiably offer exponential benefits in risk reduction at only linear cost, as discussed in block 2A.

However, I also feel that investing in greater transparency through a public bulletin board (b) will provide a similarly-valuable benefit, if less easy-to-quantify, benefit in real security, but also may be *more* valuable and effective in improving public trust in the E-voting system. Even if PB transparency mechanisms will always be imperfect implementations of an "ideal" conceptual bulletin board abstraction, and even if the "fully-untrusted PB" threat model will always necessarily rely on assumptions about side-band communication paths and voter checking behavior, nevertheless the "bulletin board" metaphor and "independent verification" capability are highly useful and important. I believe that incorporating a reasonable PB into the design will provide a major improvement in both real and perceived transparency and hence must be undertaken.

Caveat: Of course I'm well aware that I have been arguing (and will continue to argue) that "all of" quite a few nontrivial, likely expensive and time-consuming design enhancements should be made to the E-voting system. Without retreating from these positions, I also would like to make sure they're considered in the context of my over-arching position on the importance of the E-voting program in general, and in particular my position that these time-consuming improvements need not and probably should not represent barriers to the near-term deployment of an adequately-validated implementation of the current design to serve limited voter populations, despite the current design's limitations. The long-term risk of losing forward momentum is a much worse systemic risk than all the individual short-term risks of these various design improvements not yet being completed.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for contributing to this discussion. It's time to wrap it up, so here is my attempt at a summary.

Recalling Block 2, would you recommend Bob to introduce a PB or rather to procure new control-component software or adapt the protocol to achieve parameter generation that is more trustworthy, assuming he could only pick one?

Few experts were willing to address the hypothetical choice between more software diversity, more trustworthy parameter generation and the introduction of a Public Bulletin Board. One could conclude that it is a very hard choice or that it is a very hypothetical one.

Software or more generally implementation diversity brings a quantifiable benefit in security with linear costs. As discussed in 4A und 4B, a Public Bulletin Board can be a valuable element of an internet voting system which can establish trust.

If you do not agree with my summary, then please shout. If you think the *Last Call* is the moment where the discussion really starts, be my guest. The summary is very thin so far, so it would certainly profit from more opinions. What I wrote above, is a minimum summary that does not really address the question.

In case there is none or no negative feedback I will sooner or later assume this minimal consensus and close this discussion.

[EDIT] Removing some redundancy from this summary here on request of @barbara.erni after it was put back on the table in 4A.

Edited by Christian Folini 4 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Srdjan Capkun @Srdjan.Capkun · 5 months ago Developer

I think this is a good summary.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Sounds good to me as well!

Christian Folini @christian.folini · 5 months ago Maintainer

We have not heard any additional comments, so I am closing this discussion with my summary above as the conclusion.

Thank you everybody for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Discussion Block 5 - Examinations Mandated by Government

1. Introduction

The goal of the examinations is to verify that all the requirements set forth in the ordinance have been met. For example, this includes auditing the source code or reviewing voting procedures.

In this block we only discuss the examinations that are officially mandated. According to responses to question 3.3, the entity mandating the examinations should be the federal administration (often referred as the government in the responses) or an independent committee. Additionally there can be examinations run by the community, based on the publicly available information. These will be discussed in another block.

This block is based on responses to questions 3.1 and 3.3, regarding independent examinations. It also includes topics from question 2.1.8 regarding standards and methods for developing and deploying.

According to the responses, three topics seem interesting to discuss: the scope of the examination of the control components, the standards that could be used, and possible action if issues are discovered during an examination.

2. Breaking down VEleS Scope 4 (Software Examination)

The VEleS defines different scopes that must be thoroughly examined.

The current scopes defined by the VEleS (Art 7) and its annex are:

Scope 1 - Examination of Cryptographic Protocol and proofs
Scope 2 - Examination of functionality (software other than control components)
Scope 3 - Examination of infrastructure
Scope 4 - Examination of Control Components
Scope 5 - Examination of protection against infiltration (intrusion test)
Scope 6 - Examination of the print office (absence of leak during printing)

According to the answers in the questionnaire it seems that the existing examination of the software implementing the cryptographic protocol should be broken into two scopes. The main reasons are differences in the required skillsets, the methodology and the available standards.

One scope would be related to the correct implementation of the cryptographic protocol and the other related to generic software engineering. The software implementing the protocol is in Scope 4, which would now have two parts.

Scope 4.a Examination of correct implementation (alignment)

The goal of this examination is to verify the alignment of the different levels of abstraction. We consider the formal description of the protocol (used for the mathematical proofs), the specification of the system (used for implementing the system) and the resulting code itself.

This scope would cover the following:

- Verify that the specification used to write the code correctly reflects the protocol defined in the formal description used to prove the security properties of the protocol. Analyze the impact of any elements that are added by the specification.
- Verify that the code correctly implements the specification. Analyze the impact of elements that are added by the code.
- Give an assessment of whether the code and the specification is written in a way that facilitates the examination of the correct implementation.

Qualifications for Scope 4.a

Experts in charge of this examination should be cryptographers, ideally the ones implied in the examination of the protocol. As discussed in the discussion of block 1, there would be the need for at least two experts.

Standards for Scope 4.a

No standards have been mentioned for this type of examination. Current research aims at generating the implementation automatically from the high level description, which would guarantee a complete alignment. It is not available yet.

Scope 4.b Examination of programming and deployment practices

This scope focuses on how the code is created and deployed.

- Examine the code and documentation that have been produced.
- Verify that the code and documentation is developed with a method that is demonstrably effective and auditable for developing secure code.
- Verify that the code is also deployed according to an demonstrably effective and auditable method for deployment of secure systems.

The first item is already in the VEleS. The other two are not explicitly part of VEleS. According the responses to question 2.1.8, this would typically include examination of the build and deployment process, usage of different compilers, code signing or the analysis of dependencies.

Qualifications for Scope 4.b

Examiners for this scope should be experts in software development for secure systems.

Standards for Scope 4.b

In the VEleS the examination of the software (scope 2 and scope 4) is based on the Security Functional Requirements (SFR) of the Common Criteria (CC) Protection Profile (PP) for voting systems. Higher EAL levels are required for the control components as opposed to the untrusted server. While it refers to the requirements of the Common Criteria, this examination is not a formal Common Criteria certification. Other standards that were mentioned in the answers to 2.1.8 are the Microsoft Software Development Lifecycle, FIPS, OWASP Application Security Verification Standard and the Voluntary Voting System Guidelines (VVSG).

FIPS is already mentioned in 3.3.6 of the annex as one of the standards for secure algorithms and key lengths.

The answers to the questionnaire indicate that a formal certification according to a standard would not be interesting in the case of the examination of the software.

3. Related Questions

The related questions are labelled Block-5.

3.1 Individual links to related questions

- Block 5 Discussion A - Scopes
- Block 5 Discussion B - Standards
- Block 5 Discussion C - Handling of non-conformity

4. Questionnaire

The thesis is based on the question 2.1.8, 3.1 and 3.3 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.1.8	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
3.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
3.3	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

6. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 5A - Scopes (Block 5 - Examinations Mandated by Government)

Reference to originating discussion block

Block 5 - Examinations Mandated by Government

Question

Looking at the 6 scopes defined in the VEleS (listed in paragraph 2) is there a scope of examination that is missing?

Do you agree with the separation of scope 4 into two different parts, to add an emphasis on examining the correct implementation of the protocol by cryptographers and to extend the scope with programming and deployment practices?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to May 01, 2021 6 months ago

Christian Folini @christian.folini added Block-5 label 6 months ago

David Basin @David.Basin · 6 months ago Developer

The difference scopes seem reasonable. Some thoughts on possible omissions or points that could be made clearer:

- 4a and 4b are important. But isn't something similar required for the non-control components in Scope 2? Don't we also want to check the code for the untrusted server, which is also implementing part of the protocol? And the parts of the protocol running on other untrusted components? As a concrete example, wouldn't we want to check that user authentication is properly implemented (protocol is correct, design specification conforms to the protocol, implementation conforms to the design)?
- Should usability be checked?
- For the print office privacy is explicitly mentioned. What about integrity?

To be clear: in the ideal case any malfunctioning would be detected by control components. Nevertheless one would still want the other components to be developed and validated with similar care to make their compromise and misuse as difficult as possible.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I'm glad to see the explicit separation of programming and deployment practices, but I feel that "deployment practices" needs to be elaborated. Most of the software attributes mentioned in both 4a and 4b are so-called "internal" attributes, i.e., of the software itself, but there are many important external attributes that can only be assessed in an execution context, such as usability and performance. I think a few more words are needed here to make this clear.

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

From a general point of view, I do not feel comfortable breaking down the scope into 6 distinct parts without having a common control framework covering those 6 areas. Such a silo approach may lead to missing threat scenarios that involve several "scopes". It is necessary to have a global consolidated view of the e-voting system security. The numbering used (4a, 4b) suggests that the "Examination of correct implementation" and "Examination of programming and deployment practices" activities only cover control components. It does not seem logical to me that the examination scope for control components be more detailed than the scope for other components. Do I miss something?

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

Regarding 4a and 4b being limited to control components (@David.Basin and @Stephane.Adamiste):

Examination of programming and deployment practices would certainly also be beneficial to the non-control component parts (scope 2).

Examination of correct implementation (of the cryptographic protocol) only applies to parts that actually implement the protocol. As David noted, the CCs make sure that any malfunctioning would be detected. This is why they are examined with a higher standard. It certainly wouldn't hurt to apply this standard to other parts of the protocol, even if they are run by untrusted parties.

Which additional parts do you think would have to be examined by cryptographers for correct implementation of the protocol?

Then, there is the question of cryptography which is not in the formal description of the crypto protocol. As David mentions, Svote has an authentication mechanism which is not described in detail in the formal crypto proofs but is found in the software specification. This is what we meant with "Analyze the impact of any elements that are added by the specification" in scope 4a.

We would expect an assessment of whether this added crypto may have an impact on the proven properties of the protocol. Should we ask for more?

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

@philippe.oechslin : I had scope 2 in mind.

David Basin @David.Basin · 5 months ago Developer

@philippe.oechslin notes that it probably wouldn't hurt to apply this standard to other parts of the protocol even if they are run by untrusted parties.

I would state it more strongly than "it probably wouldn't hurt". I would insist on it. It is all well and good that the control components will (if designed properly) detect the malfunctioning of the other components. But this detection (depending on the details of the design and what is compromised) can result in the election being aborted.

For example, if an attacker successfully attacks the server then we will be in a situation where the control components will notice problems and abort the election. This is bad. It is also bad if it leaks out to the public that the design or implementation of non-control components, like the server or voting clients, are flawed. Imagine the publicity (newspapers, social media, ...) that will result if a hacker can break user authentication and start the voting process as a voter, even if he can't do anything very meaningful. Who is actually right will be secondary once the twitterstorm starts.

Aside from the above mischief, defense in depth is a good thing in practice. This is the case even if you model the server as untrusted for the purposes of verification. To be clear about this: modeling the server as untrusted is useful for verification where it can model a situation where (in practice) we trust the authority running the server, but the server can be compromised by an attacker. Hence we would like a verified design and implementation for the server too. (A similar story can be told for the voting clients even though they really cannot be trusted.)

Said another way: it seems unusual to take less care in ensuring the security of the main parts of the system simply because there are other components around that will tell you that you have been compromised.

Reto Koenig @Reto.Koenig · 5 months ago Developer

Splitting the examinations into different scopes has its drawbacks. We could observe that during the examination / certification process from 2016 to 2019 for sVote. There, there were different groups of experts covering different scopes not paying attention to the 'big-picture'. This opened crucial gaps between the scopes that each group of experts refused to examine (implicitly / explicitly).

So splitting the system up in even more scopes will not mend that particular problem.

In our opinion, what is needed beside the different scopes is a group of experts keeping the big-picture in mind, *managing all the different transitions* between the scopes. And this, of course cannot be any certifying party, as we are not aware of any such party, that has this special knowledge in remote e-voting. So, it must be some (non-closed) group of dedicated experts able to operate transparently and independent of any 'vendor' interest . This group could then provide the final verdict of the system in terms of the big-picture.

David Basin @David.Basin · 5 months ago Developer

Concerning @Reto.Koenig 's statement: "This opened crucial gaps between the scopes that each group of experts refused to examine (implicitly / explicitly)." Groups were aware of the gaps and they were explicitly highlighted. It should be clear to all: one cannot look at proofs independently of design or design independently of implementation. The gaps must also be analyzed.

In any case, I am in complete, emphatic agreement with the last paragraph: there is a need to have a group doing a holistic security analysis and looking at the big picture and this group should work transparently and independently of any vendor interest.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

The scope makes sense to me. However, I also would like to support the point that @philippe.oechslin, @Reto.Koenig, and @David.Basin were making. By focusing on scopes and staying in silos, or, by staying at a too high level of abstraction, certain threats fall in between the cracks. We speak a lot about reviewing designs and analyzing implementations, but deployment plans are often ignored. This is problematic. Just look at NSW in 2015: Vanessa and Alex Halderman managed to inject malicious Javascript code into the voting client from a poorly secured third party server. So a holistic review should also involve deployment plans.

Edited by Carsten Schuermann 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

@Carsten.Schuermann minor nitpick: I can't parse your sentence "The scope makes to me." I guess you meant "The scope makes sense to me"?

Otherwise, I agree with all the main points brought up above and can think of nothing substantial to add at the moment.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Thanks @Bryan.Ford

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for participating in this discussion. Let me try to sum it up.

Looking at the 6 scopes defined in the VEleS (listed in paragraph 2) is there a scope of examination that is missing?

Do you agree with the separation of scope 4 into two different parts, to add an emphasis on examining the correct implementation of the protocol by cryptographers and to extend the scope with programming and deployment practices?

The outlined scopes make sense. Yet applying this high level of scrutiny only to scope 4 and thus the Control Components is seen as an unnecessary prioritization. A successful attack on the other voting servers (scope 2 and 3) could severely undermine the confidence into the system regardless whether a control component detected a manipulation or not.

The examinations must also be conceived and performed from a holistic view. Performing the examinations only within the individual scopes without taking into account the full context could lead to important aspects being missed. A way to cover this would be a group of experts that is in charge of managing all the different transitions between the scopes.

Usability and performance have been mentioned as examples of external attributes of the software that should be taken up in the examination framework.

The decision on the assurance standard should not be determined by whether a component is declared trusted / untrusted in some model. As an example: Although untrusted servers that misbehave at authentication would be detected thanks to a trusted group of control components, failures in the authentication process are still bad and avoiding them deserves focus.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

There have not been any requests to adjust the summary. I'm thus closing this discussion. Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Discussion 5B - Standards (Block 5 - Examinations Mandated by Government)

Reference to originating discussion block

Block 5 - Examinations Mandated by Government

Question

Scope 2 (examination of functionality) and 4 (examination of control component) are based on elements of the Common Criteria e-voting Protection Profile and Assurance Levels 2 and 4 respectively.

Scope 3 (infrastructure) requires an ISO 27001 certification.

Scope 5 (penetration test) mentions OWASP.

Do you see aspects that could be borrowed from other standards and would be beneficial to the examinations?

Do you see a value in requiring a formal ISO 27001 certification as opposed to an examination based on that standard?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to May 02, 2021 6 months ago

Christian Folini @christian.folini added Block-5 label 6 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

As already mentioned, standard SE and process standards would also be relevant for Scope 4b (i.e., ISO 9000 and friends).

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

In my opinion, the way this question is phrased does not reflect the original question: "Do you know any standard that would likely lead to better procedures at development?". Now, we are only focussing on examinations, skipping secure development practices that should be adopted by software development teams.

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

The examination is also meant to examine the development, not only the final product. Sorry if that wasn't clear. If we have a standard for better development, we would want to examine whether the standard was followed.

David Basin @David.Basin · 5 months ago Developer

As noted in my comment to 5A, although I understand the logic for treating control components as more critical than other "functional" components, I think having weaker assurance standards for the functional components would be a mistake (for the reasons explained in 5A).

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I would suggest following the good practices set by the Microsoft Secure development Lifecycle framework to streamline security activities during software development. It is not a standard though, but a comprehensive set of activities to be performed at the various phases of a development project. Chapter 14 of the ISO27002 standard (system acquisition, development and maintenance) may be used as a reference to examine software development practices. It includes 12 controls and is meant to have

a comprehensive coverage of security aspects within a software development project.

From a general point of view, it would make sense to follow a common security assurance framework for the whole e-voting scope. In other words, I do not understand the logic behind the fact that infrastructure should be ISO27001 certified and not the other blocks that form the e-voting system (software development and operation processes). The necessary holistic approach to information security to efficiently reduce risks would be better enforced if all blocks were considered homogeneously.

The value of requiring a formal ISO27001 certification lies in the fact that it would force the certified body to apply a continuous improvement approach for information security management. The latter would have to audit itself on a regular basis, thus improving the examination process.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I think the NIST Cybersecurity Framework might be of interest as well especially for Scope 3.

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

@Stephane.Adamiste The issue with ISO 27002 chap 14 is that it is very subjective. Typical wording is "the following aspects should be considered". How far you actually go for each aspect depends on a risk assessment and is highly subjective.

Common Criteria works the other way around. You first declare the level of assurance you want to reach. Then you have no leeway. You must implement the listed functional requirements (e.g. logging with timestamps) and assurance requirements (eg. analyze the coverage of your tests, provide complete functional specification).

The examination of the software is a basis for authorizing the use of the system. Thus the standard used should be as objective as possible. The provider needs to know what it will be measured against and the regulator needs objective arguments for authorizing/refusing the system.

Do you think this could be achieved with ISO 27002 ?

Collapse replies

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

@philippe.oechslin In my opinion, a suitable way of proceeding is to link Common Criteria to related ISO27002 controls. Functional requirements could be used as an input for ISO 27002 14.2.9 System acceptance testing (Acceptance testing programs and related criteria should be established for new information systems, upgrades and new versions).

[edit] Improved text following @David.Basin 's remark below.

Edited by Stephane Adamiste 5 months ago

David Basin @David.Basin · 5 months ago Developer

Maybe I misunderstood @Stephane.Adamiste suggestion, but a Common Criteria Certification seems like an awful lot to shoe-horn into ISO 27002 14.2.9. E.g., CC has assurance requirements including (to name a few): configuration management, secure delivery, development requirements (Functional specification, high-level design, low-level design, implementation representation of ToE security functions, representation correspondence, security policy modeling, ...), Life Cycle Support, Testing, Vulnerability Assessment, etc. This goes far beyond acceptance testing.

Collapse replies

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

@David.Basin , my mistake. I mixed "functional requirements" with "Common Criteria" in my previous sentence. I have rephrased my comment. Hope it makes more sense now.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for participating in this discussion. Let me try to sum it up.

Scope 2 (examination of functionality) and 4 (examination of control component) are based on elements of the Common Criteria e-voting Protection Profile and Assurance Levels 2 and 4 respectively.

Scope 3 (infrastructure) requires an ISO 27001 certification.

Scope 5 (penetration test) mentions OWASP.

Do you see aspects that could be borrowed from other standards and would be beneficial to the examinations?

Do you see a value in requiring a formal ISO 27001 certification as opposed to an examination based on that standard?

ISO 27002 seems to be broad enough to cover all the scopes of the examinations, which is desirable, but it comes with some limitations. Its risk based approach and its lack of detail in the domain of security and assurance requirements make it a poor choice for examination of the software.

The following frameworks/standards were mentioned as foundations for scope 4b: ISO 9000, Microsoft Secure development Lifecycle framework, ISO 27001 and ISO 27002 as well as Common Criteria.

Standards for security and privacy are under active development with new standards emerging regularly. It is important to keep a close eye on this area.

For scope 3 the NIST Cybersecurity Framework was mentioned.

There is a certain danger, that the disparity of control frameworks for the different scopes might cause a lack of consolidated view on the security of the system as a whole.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will assume consensus and close this discussion towards the end of the week.

[EDIT]: Updated on request of @Stephane.Adamiste below.

[EDIT]: Added paragraph "Standards for security" based on a suggestion by @Carsten.Schuermann.

Edited by Christian Folini 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I think one issue was raised that does not appear in the summary: Why this heterogeneity in the references for examination? What are the rationales behind the selection of a given framework/standard for a given scope? Is it not a problem to get a consistent consolidated view of the system's security posture as a whole?

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your comment, @Stephane.Adamiste.

The summary is very sparse, since we only got a few responses in this thread.

Could we prepend the following sentence to address your concerns?

> ISO 27002 seems to be broad enough to cover all the scopes of the examinations, which is desirable, but it comes with some limitations. Its risk based approach and its lack of detail in the domain of security and assurance requirements make it a poor choice for examination of the software.

Alternatively: You can also propose a wording that addresses the other points that you think should enter the summary.

Collapse replies

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

Sounds good. Could we underline also that the disparity of control frameworks for the different scopes might cause a lack of consolidated view on the system's security posture as a whole?

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Stéphane.

I think your subsequent addition makes sense. I have added it to the summary and wait for potential feedback before closing this.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Can we add a sentence to the effect that "standards for security and privacy are currently actively under development with new standards emerging (for example 27400 series IoT security and privacy), so it is important to keep a close eye on the state of the art".

Christian Folini @christian.folini · 5 months ago Maintainer

This seems to make sense.

I suggest we word it as follows and place it before the *For scope 3*:

> Standards for security and privacy are under active development with new standards emerging regularly. It is important to keep a close eye on this area.

(I'm not referring to the 27400 series as people would ask, how they relate to internet voting.)

@Oscar.Nierstrasz: Is that what you had in mind?

@Stephane.Adamiste, @David.Basin, @philippe.oechslin, @Bryan.Ford, @Oscar.Nierstrasz: I reckon you are OK with this amendment. If I do not hear from you, I'm going to update the summary.

Christian Folini @christian.folini · 5 months ago Maintainer

I see 3 upvotes for the proposal and no other feedback. I have thus added the new passage and I am closing this discussion now.

Thank you for contributing to this discussion.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 6 months ago by Christian Folini Reopen issue

Discussion 5C - Handling of non-conformity (Block 5 - Examinations Mandated by Government)

Reference to originating discussion block

Block 5 - Examinations Mandated by Government

Question

Imagine that non-conformities have been detected in one of the examinations and that an independent entity has run a risk analysis to identify the risk created by these non- conformities.

In ISO 27001 certification, for example, it is often the case that a certification is given even if some less critical conformities have been detected.

Would you agree that the system could still be used if the analysis shows that the risk is low enough?

Consider that if zero risk is required, then any small issue could prevent the system from being used.

Do you think that some scopes are more critical than others? If that is the case, do you see any scope for which no non-conformity should be tolerated?

Here are some typical examples of issues that could be discovered

- the code does not exactly implement the protocol, but there does not seem to be an impact.
- the development method does not conform in every aspect to an effective and auditable method, but the code is correct.
- the penetration test shows some missing best practices, but there is no exploitable vulnerability.
- the ISO audit shows a lack of documentation, which will only be fixed in 6 months.
- the examiners declare that they did not have enough time to complete the examination.

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to May 03, 2021 6 months ago

Christian Folini @christian.folini added Block-5 label 6 months ago

David Basin @David.Basin · 6 months ago Developer

Not all nonconformities are the same.

Here are things that must be fixed:

- If a protocol has a design error, this must be fixed as it would lead to an attack.
- If the protocol description (used for verification) and the design specification (used for implementation) are not conform then this must be fixed as otherwise the formal analysis says nothing about the actual system implemented.
- If something different was implemented then described in the design specification or if there are bugs at the implementation level, then these must clearly also be fixed.

Note that all of the above are focused on THE PRODUCT, i.e., the system and its design. Some standards, like ISO 27001 are focused more on THE PROCESSES around the product, e.g., security management. Having good processes and following best practices is a good idea, as it tends to lead to better products. However it is neither necessary nor sufficient for having a secure product. So of the "typical examples" above, I would allow for some flexibility for process deviations provided that the product is secure, e.g (from list above) the code is correct, no exploitable vulnerability, etc.

Note too that part of making the product secure is having a hardened system, as some software bugs (that escape review) are inevitable. Hence part of the product (system) includes firewalls, PKI setup with certificates for authenticating components, patch-level, etc. There should be very little or no flexibility here as defense in depth will play an important role in the overall security, in practice.

Edited by David Basin 6 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

Just to clarify, ISO27001 auditors certify that a given organisation has the organisational and technical means in place to manage information security aspects related to a given perimeter by implementing a Plan-Do-Check-Act approach (continuous improvement). Running an Information Security Management System does not mean that all risks are at a residual acceptable level (btw, "zero risk" does not exist, we should avoid using such a misleading expression). ISO27001 is about defining security requirements for information assets and applying an iterative approach to ensure that risks pertaining to those information assets are treated so that they are acceptable for the management (risk acceptance being a way to treat risk).

Edited by Stephane Adamiste 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

Would you agree that the system could still be used if the analysis shows that the risk is low enough?

Are there any alternative? We all know, zero risk does not exist, don't we? The big question in information security management is to define what "low enough" means in terms of risks relating to a given system. For instance, one could reasonably think that risk inherent to postal voting, which do not seem to be questioned by the public opinion, should also be accepted for e-voting. This could be one kind of threshold for risk acceptance. Also, if a threat agent is able to reach his/her/its goal in a far easier way than compromising information processed in an e-voting system, there is little probability that he/she/it loses time trying to hack into a system that has proven to be robust. Performing intelligence operations on Facebook has shown good results in the Brexit case at a reasonably low cost. No need to hack into an e-voting system...This kind of considerations should drive the effort put on the e-voting security effort.
Coming back to the vulnerability examples quoted, the risk management methodolgy applied to the e-voting system should show how those vulnerabilites impact risks. This is a tricky part.

Edited by Stephane Adamiste 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

This is a good question, but the problem more complex than the formulation suggests. The decision taken in response to observing a non-conformity depends on when the observation is made and how severe it is. (I think this is clearer than talking about risk).

If the non-conformity is severe and there is time (even if only hours), then, most likely, it will be fixed even though the election is in progress. Examples, where this has happened in the past: In the Parliament election, Norway 2013, there was a problem with randomness in the voter client. They fixed the voting client half way through the election, although the official statement read that enough operational security was put in place to protect vote privacy. One could ask then, if the operational security was good enough, why did they risk fixing the system in flight? Another example is NSW 2015, when the system was updated to prevent third party JavaScript injections.

Sometimes, non-conformities cannot be determined until it is too late (Presidential election, Kenya 2017, results-transmissions problem with KIEMS devices), which can have severe consequences, in the case of Kenya, the Supreme Court ordered a Fresh Presidential election.

Therefore, when building any kind of Internet Voting system, it is important to be realistic. Certification, security analysis, penetration testing will not reveal all non-conformities. Therefore, it is critical to (1) allow for enough time to identify as many non-conformities as possible and (2) put mechanisms in place to respond to non-conformities detected last-minute or during the election.

Collapse replies

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I agree that talking about non-conformities is clearer than talking about risk for some issues, typically the ones that can be fixed (e.g. a flaw in the code allowing javascript injection, as you mentioned). But in other circumstances, I hardly see how we could skip the residual risk evaluation and acceptance process. The case of a library with known vulnerabilities for instance: There is a trade-off to make between system stability and keeping the versions up-to-date in terms of security.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Fully agreed that "zero risk does not exist."

In evaluating whether a discovered non-conformity can or should be tolerated, this is where it is essential not just to consider in isolation the risk of using (or not using) an imperfect E-voting system, but also to consider the risks involved in *not* using it and the comparative risks of the available "baseline" alternative(s).

In the Swiss context, in particular, some risks of choosing *not* to use an E-voting system in which a non-conformity has been detected may include:

- Effective disenfranchisement of overseas Swiss voters for whom round-trip postal voting is too slow.
- Exclusive "monocultural" reliance on a postal voting system that holds many less-obvious risks of its own, as explored by Killer/Stiller for example (see discussion 99A).
- The long-term systemic risk of losing forward momentum toward (and funding for) more secure alternatives to all of the imperfect current voting mechanisms, and hence remaining technologically "stuck" and unable to respond to increasing threats against the predominant postal voting system for example, as I discussed in 99A.

There are no easy or clear-cut answers here, unfortunately, only complex and difficult balances among many unknowns. But the implicit assumption that I think many people including technology experts often make, that not using E-voting is always unconditionally safer than using E-voting no matter what, is overly simplistic.

Edited by Bryan Ford 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for participating in this discussion. Let me try to sum it up.

Imagine that non-conformities have been detected in one of the examinations and that an independent entity has run a risk analysis to identify the risk created by these non- conformities.

In ISO 27001 certification, for example, it is often the case that a certification is given even if some less critical conformities have been detected.

Would you agree that the system could still be used if the analysis shows that the risk is low enough?

Consider that if zero risk is required, then any small issue could prevent the system from being used.

Do you think that some scopes are more critical than others? If that is the case, do you see any scope for which no non-conformity should be tolerated?

Here are some typical examples of issues that could be discovered

- *the code does not exactly implement the protocol, but there does not seem to be an impact.*
- *the development method does not conform in every aspect to an effective and auditable method, but the code is correct.*
- *the penetration test shows some missing best practices, but there is no exploitable vulnerability.*
- *the ISO audit shows a lack of documentation, which will only be fixed in 6 months.*
- *the examiners declare that they did not have enough time to complete the examination.*

Based on the assessment of “risk” or “criticality”, it can be reasonable to accept non-conformities. However, the risk itself can be hard to assess.

With regards to the risk, points of reference might be found in postal voting or in other systems that would likely allow a threat-agent to reach his goal to manipulate the outcome of a vote at a lower cost. The risks of not offering internet voting may also be taken into the equation, e.g. disenfranchisement of voters abroad or losing momentum and resources towards more secure solutions, thereby possibly also addressing the risks inherent to postal voting.

Situations where non-conformities become subject to acceptance should be avoided by allowing enough time for detecting / fixing them. Mechanisms should be in place to respond to non-conformities detected at the last-minute or during the election.

Generally, the more the product (including the operating infrastructure, e.g. firewalls, PKI, ...) is affected, rather than the processes around the product, the more important it is to fix a non-conformity.

The following types of non-conformities have to be fixed:

- If a protocol has a design error, this must be fixed as it would lead to an attack.
- If the protocol description (used for verification) and the design specification (used for implementation) are not conform then this must be fixed as otherwise the formal analysis says nothing about the actual system implemented.
- If something else has been implemented than described in the design specification, then this must clearly be fixed.

- If there are bugs on the implementation level, then they must be fixed too.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation. In case there is no feedback or no negative one, I will assume consensus and close this discussion towards the end of the week.

Christian Folini @christian.folini · 5 months ago Maintainer

There have not been any requests to adjust the summary. I am thus closing this discussion. Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

6-development-and-publication.md 20.6 KB

Discussion Block 6 - Development and Publication

1. Introduction

The existing regulation mandates the publication of the source code after the 100% certification (VEleS article 7a, 7b). The idea is to gain transparency, a higher level of trust and also grow the number of people who are familiar with the system and online voting technology in general. It should also allow authorities to make early improvements in case errors are found. The publication of the source code of the Swiss Post / Scytl internet voting system combined with the public intrusion test led to global media attention. Several highly critical findings were identified in the source code.

The modalities of the publication of the source code in 2019 were criticized in the media and by politicians alike. Also the experts expressed strong criticism with the source code and its publication in the responses to the questionnaire: The source code and its documentation should to be more accessible, easier to read and easier to audit. The researchers should be able to compile the code, and they should be able to conduct a complete ballot on their own.

There is no agreement on the level and the extent of the proper transparency. Likewise, we don't know yet how to achieve continuing scrutiny.

In their answers to the questionnaire, some experts opt for the adoption of an open source license for all the code developed for the internet voting systems. A larger group of experts just wants to make sure everybody gets access to every part of the code and documentation needed to audit the system from various directions, possibly even on an on-demand base for some parts of the documentation.

The primary goal of the transparency measures is valuable scrutiny for the system. So far, the regulation only expects the publication of the source code of the finished system. For some experts this does not seem to be sufficient. They would also like to see the evloution of the system. There is thus a discussion if scrutiny would profit from regulating the development process in the direction of more openness and transparency. Furthermore, the rights and duties of the public with regard to the usage of the code (license choice) could have a potential to impact the obtained scrutiny as well.

During the years, the Geneva internet voting system published more and more of its source code with an open source license and adopted a more and more open development practice. However, there was very little contribution from 3rd party developers hitherto.

The Swiss Post / Scytl system has been developed in a closed mode so far and the publication led to numerous researchers examining the code after its publication. This may or may not have to do with the public intrusion test carried out in parallel (Also see block 7 for a discussion of the public intrusion test). However, an extensive publication of the source code doesn't guarantee that researchers and third party developers continually participate in the improvement of the code.

There is very little precedent with regulating publication of source code and system documentation in Switzerland. However, the new extension of the Epidemics Act contains the following passage on the Proximity-Tracing App: “The source code and technical specifications of all components of the PT-System are public. The machine-readable programs must have been demonstrably created from this source code.” (Epidemics Act, article 60, section 5e, change of June 19, 2020; inofficial translation of the official German text)

2. Related Questions

The related questions are labelled Block-6

2.1 Individual links to related questions

- Block 6 Discussion A - Scope and quality of the publication
- Block 6 Discussion B - Development mode as a means for transparency and public scrutiny
- Block 6 Discussion C - The point in time of the publication
- Block 6 Discussion D - Assessing the level of public scrutiny and trust

3. Questionnaire

This block is based on the answers to questions 4.1, 4.2, 4.3, 4.4 and also touches on some aspects of 4.11.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
4.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
4.2	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.3	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.4	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.11	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

4. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 6A - Scope and quality of the publication (Block 6 - Development and Publication)

Reference to originating discussion block

Block 6 - Development and Publication

Questions

We assume the publication includes the source code of the voting systems, of the client code, server code, control components, and of the verifier.

VEleS 7a para. 3 excludes the following items from the obligation of publication: Source code of freely available and regularly updated systems that are in widespread use like OS, DBs, application servers etc. Is this reasonable?

Which parts of the documentation have to be published (please also think about specifications, namely the discussion 1C. Furthermore, would it be acceptable if some documents would only be made available on demand)?

If you think about setting up and conducting a complete ballot on your own: What changes if the test data and parameterization is published or not?

How big is the need to publish audit documents (including certification and penetration tests mandated by the system provider)?

What would be the essential reasons to ask for documentation and configuration of auxiliary systems and peripheral security measures (e.g. web application firewall configuration, processes to obtain administrative super-user access on production servers, system hardening information, patch levels, etc.)?

What would be your requirements if you had to decide if the publication of the source code, documentation, etc. was acceptable in terms of quality, readability and auditability?

Is a registration or an NDA acceptable to access the code and the documentation?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to June 01, 2021 5 months ago

Christian Folini @christian.folini added Block-6 label 5 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Publication should include all developed source code (including tests) and documentation, i.e., everything that would be needed to install and maintain the system. It excludes source code of any subsystems and infrastructure provided by any third parties (O/S, DB, GUI etc). A fundamental question is if the system will be closed or open source. This impacts the answers to all the other questions (NDA etc). In any case, parts of the system (code, documentation, logs, audits, issues etc. that are kept secret, will make it harder to find and expose vulnerabilities lurking there. Either make everything open and public, or keep everything secret. I don't see the sense of taking a middle road.

Concerning quality standards, it is a fundamental task of project management to establish and maintain these throughout the project. They are subject to monitoring and control in any serious project. They are always established based on the needs of an individual project, and cannot be fixed a priori.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for this first response in this discussion block.

You are talking about closed vs. open source, thus the license question (which is likely to apply to question 6B. But then you also

talk about public vs. secret.

I'd like to be sure your position becomes really clear: Can closed source development become open and public via a publication (the way the regulation demands it now)? Or is this model part of the middle road which you think is not useful and it should better remain secret then?

I think that would be a possible position, but it is a new position and I think it ought to me highlighted as such.

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

AFAIK, the path from closed to open source is quite common, and I have seen some cases. There is nothing wrong with it.

David Basin @David.Basin · 5 months ago Developer

I agree with @Oscar.Nierstrasz: If the source code is published then there is no reason not to go all the way and include everything needed to build and run the system.

I would also publish documentation and specifications. But audit documents are less clear: there may be some iteration involved. How much of this history do you wish to expose? Will publishing it all (including all the issues dealt with) increase trust? I have my doubts.

Patch level should be clear from code distribution, which is published.

Hardening measures are by nature somewhat heuristic. I would be satisfied if this information was audited by a competent auditor, including being subjected to pen-tests, without full publication of every last detail. The same holds for firewall configurations.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thanks David.

It's a detail here, but just to be clear: Patch level means that the system provider publishes the patch level of all involved systems. This not what the OS vendor says they should be patching, but what package version they actually have on the individual production servers. This is a requirement that would be in line with the idea of *maximum transparency* and we want to hear from you experts what this boils down to with very concrete examples.

So if you tell us, that you want a system provider to prove to the public they are able to patch production within 24h whatever happens (and everybody can run the stop watch), then this would be a strong statement.

But if you tell us, this is not necessary and it is OK to have a strict policy in place to patch within 24h and actual patch levels should not be shared because it would be a major information leak, then that is an equally strong statement.

Florian Egloff @Florian.Egloff · 5 months ago Developer

you asked: How big is the need to publish audit documents (including certification and penetration tests mandated by the system provider)? I think: all the audits and certifications should be publicly available as they give indications to which parts of the code-base was previously found lacking in security. One ought to assume that a competent adversary has access to them and thus give access to them to those independently testing the security of the system.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Public vs closed code. Well, in my opinion, the code should be public. Building on closed source code makes no sense. The tools for reverse engineering have become very good. I just reversed the binaries of the AutoMark machine (old, but still in use in the US 2020 election) and with dotPeek, you recover all of the source code (including all hard-coded passwords, but no comments). It is incredible how good these tools have become. So you might think that since binaries run on some server behind some firewalls, they are out of reach to the adversary. I don't believe that is a viable argument. If there is a will, there is way. Given enough interest, those who want to get access to the code will be able to.

Balance between transparency and opaqueness. Every country seeks to find a balance between transparency and opaqueness of its operation. Within this compromise, the best course of action is to be as transparent as possible. The source code represents the instructions on how how to run election day activities, this should be public. When developing source code, publishing intermediate

commits is optional, whoever, I would argue that also they should be made public as they contain meta-data that can be trust-building: it shows how many people worked on the project, when most of the development activities took place or what has been done to the source code since the results of the PIT became public.

Public participation in creating, securing, reviewing, and maintaining the code base. I think this is wishful thinking. I predict that no many would voluntarily engage in this kind of activity. People don't really want to review the code, but the want to know that others could it. This is were pubic confidence is created.

Patch levels. Of course, these must be known. How can you evaluate the engine of car, if someone removed all screws, and doesn't want to tell you which ones you have to put in and with which torque? I have tried to build many voting system over the years, not to run them, but to be able to use automatic code scanning tools on them, for example Fortify, Coverity, etc. to find common vulnerabilities. Often it is possible to reconstruct which versions of a particular library were used, but knowing the correct version numbers would simplify the whole process.

Regarding publishing auditing reports. My experience has shown, how important it is that these reports are public, not alone to give election observers the possibility to speak with confidence of the quality of a system.

Edited by Carsten Schuermann 5 months ago

Christian Folini @christian.folini mentioned in issue #40 (closed) 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Open versus closed source: I agree with the opinions above that everything developed for the E-voting system should be open source and fully reproducible and independently testable. I also agree that underlying components such as libraries, operating systems, and hardware need not be open source (and it would simply be unrealistic to expect it to be), provided those closed-source components are *standard* and widely-used.

Closed-source binary software components: In the inevitable case that some underlying software components are closed-source, an important further question is what precise *binaries* are actually used in the deployed systems? Despite being closed-source, the major operating systems like Windows and Mac OS get a lot of constant scrutiny by independent security researchers with sophisticated binary-analysis tools such as those @Carsten.Schuermann refers to above. The trustworthiness and (both real and perceived) transparency of E-voting system components running on such closed-source platforms can benefit from this independent scrutiny of the major closed-source operating systems, but *only* if the deployed E-voting systems are actually running the well-known binary versions that have received this scrutiny. How are we to know, then, that a control component (for example) is running one of the standard, well-known, public binary release versions of that OS, and not a version with (perhaps secret) patches applied or nonstandard closed-source drivers or libraries installed, which might hide either accidental vulnerabilities or maliciously-inserted back-doors?

This is where hardware security mechanisms like secure attestation (e.g., secure boot, Intel SGX) can provide useful, if imperfect, ways to increase strong confidence especially in closed-source components. And even in the open-source components, it is similarly desirable to be able to know exactly what binary is running and how it was compiled (ideally via a reproducible build), so that independent experts can verify that the correct code was built with a standard compiler toolchain and not a potentially compromised one (see again Reflections on Trusting Trust).

Patch levels: Responding to the question above by @christian.folini about patch levels: I do think it is important that the deployed system is guaranteed to be using *some* well-known and reasonable patch level of the standard system - and ideally verifiably so as discussed above.

Whether the precise patch level needs to be disclosed to the public immediately, all the time, is a separate and trickier question. On the one hand, it is important that the critical E-voting system components are being kept up-to-date with important security patches, and it is relevant to transparency arguments for the public to be able to verify that this is happening. On the other hand, in any production system, validating new OS/platform software versions against any critical application before rollout is always important to head off the risk of system breakdown/unavailability due to unexpected incompatibilities introduced by the new version, newly-exposed performance bugs, etc. When a patch newly-released by an OS vendor is known or strongly-believed not to affect anything the E-voting system depends on, it may be justifiable to delay the new patch's rollout to minimize unavailability risks, especially if an election is in progress. If a new patch does address a potentially-relevant security weakness, then there is arguably some security benefit to the attacker not knowing whether or exactly when the fix is rolled out on the production system: an attacker who doesn't know whether their hack will silently succeed or be immediately detected may be much more hesitant to attack at all than one who is sure the attack will succeed based on knowledge of the exact running version. For these reasons, I see value in leaving some operational flexibility in deciding exactly how quickly and exactly when new patches are rolled out, and in revealing or hiding exactly which of the standard, recent well-known patch levels is in operation.

Making this assurance publicly verifiable - that *one* of the recent well-known patch levels is in operation without necessarily disclosing which one - is another challenge that should probably be explored further. Ring signature and group signature type mechanisms, already mentioned in other contexts, could in principle be useful here, for example to enable a trusted-boot or hardware-attestation workflow to "prove" to the public that one of a specific "anonymity set" of recent patch level images is running without disclosing which one. A related and perhaps complementary possibility might be for such a trusted-boot process to verify and boot *only* binary images that have been witness-cosigned by some number of independent entities, as I discussed here for example. This would effectively ensure that the running image can only be one of those that have been "publicly witnessed and recorded", while leaving the choice of *which* witness-cosigned image is running at a given moment up to the choice of the operator.

Documentation and configuration of auxiliary systems: For these, the "closed versus open" question is a bit less clear to me. As long as the "auxiliary systems" we're talking about are by design supposed to be completely untrusted for E-voting system security and privacy (i.e., the voting system formally assumes that all such auxiliary systems are adversary-controlled), there may be security value to having some of this closed, or publicly released only at the discretion of the operator, in the interest of making it more difficult for a potential remote attacker to guess or figure out what exact kinds of perimeter defenses they have to get through to get to the critical components, and to make it arguably more likely that the defending operators will be able to detect such an attack attempt quickly as it happens, because the attacker will have to do more "probing" than he would with perfect knowledge of the auxiliary systems. Nevertheless, there should at least be public documentation of some *minimum* baseline set of facilities and security measures deployed, if not necessarily the full set or all the precise configuration details.

NDAs: I would strongly advise against requiring anyone to sign NDAs as a condition for getting access to the source code or any of the documentation or other materials that is supposed to be publicly verifiable. In practice, NDAs are far too often used (especially by industry) as a way to censor potential criticism or to pressure responsible, well-intentioned security researchers to delay or suppress public release of their findings. Because of this association of NDAs with censorship, among others, requiring the signing of NDAs is likely to undermine the trust of both experts and the public.

Furthermore, for the above as well as other cultural reasons, many of the brightest experts who might be willing and interested to examine the code carefully will never sign an NDA for reasons of principle. Therefore, requiring people to sign any form of NDA will undoubtably reduce the amount and breadth of valuable scrutiny the code receives by a substantial amount, no matter how innocuous and reasonable the lawyers might think the NDA's terms are.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Time is running out, so we better get going. Here is a draft summary of this question. I ask you to review it please.

We assume the publication includes the source code of the voting systems, of the client code, server code, control components, and of the verifier.

VEleS 7a para. 3 excludes the following items from the obligation of publication: Source code of freely available and regularly updated systems that are in widespread use like OS, DBs, application servers etc. Is this reasonable?

Which parts of the documentation have to be published (please also think about specifications, namely the discussion 1C. Furthermore, would it be acceptable if some documents would only be made available on demand)?

If you think about setting up and conducting a complete ballot on your own: What changes if the test data and parameterization is published or not?

How big is the need to publish audit documents (including certification and penetration tests mandated by the system provider)?

What would be the essential reasons to ask for documentation and configuration of auxiliary systems and peripheral security measures (e.g. web application firewall configuration, processes to obtain administrative super-user access on production servers, system hardening information, patch levels, etc.)?

What would be your requirements if you had to decide if the publication of the source code, documentation, etc. was acceptable in terms of quality, readability and auditability?

Is a registration or an NDA acceptable to access the code and the documentation?

All developed source code and everything needed to understand, to install and to maintain the voting system has to be published. This is meant to allow researchers or auditors to run an election or vote in their own premises with exactly the same reproducible build of the software that is used by the official system.

Any part of the system that is not published will make it harder to find and expose vulnerabilities lurking there.

Being as transparent as possible will help to build confidence towards the voting system. Not necessarily because people will actually review the system themselves, but because they know that other people with the necessary know-how and interest have the possibility to do so.

There is a strong call for the publication of audit documents as well, yet it is not unanimous.

It is acceptable to use widely used standard components such as operating systems, databases, firewalls, etc. without publishing their source code when there is evidence, that these systems have seen a lot of scrutiny elsewhere and the voting system can profit from these efforts. Technical means to enforce the use of a specific, public and widely-used build of said software would be useful.

Publishing detailed patch levels of all sub-systems as well as configuration of peripheral elements of the voting systems such as firewalls would be welcome to several experts. But there are also experts who see this as an information leak that brings an advantage to the attackers, since they are no longer forced to run extensive probes to gather this information. Forcing them to probe can reveal useful information about the attackers in return, so this practice should not be seen as a mere "security by obscurity".

Any form of NDA will massively reduce the number of qualified researchers looking at the code.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Reworded the paragraph on standard closed source components based on input by @Olivier.Pereira and a slight rewording of Olivier's proposal by @Bryan.Ford.

[EDIT] Extended the section on information leakage to make sure it can not be read as security by obscurity. On request of @Carsten.Schuermann.

Edited by Christian Folini 5 months ago

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Hi @christian.folini,

That looks all good. I am just in doubt regarding: "The source code of widely used standard components such as operating systems, databases, firewalls, etc. need not be published since they are widely scrutinized and the voting system can profit from these efforts,".

I am not sure that we can always argue that "widely used standard components" are all "widely scrutinized" and, in particular, the level of scrutiny may be hard to determine for closed-source systems, and in particular for closed boxes like firewalls? (Widely used open source may not be better -- the press talked a lot about the case of OpenSSL after Heartbleed for instance https://threatpost.com/openssl-past-present-and-future/112485/.)

It seems that either those components are open source, in which case there is no need to publish them as part of the voting system since the code is already available (and it might be good to track whether the corresponding projects actually have an active base of reviewers), or that those components are closed source, in which case it would just be illegal to publish the code (and probably hard/infeasible to just gain access to it), and the question would then be to explain why the closed-source code should be selected and trusted, which may be argued based on the approaches described in the comments above -- but not taken for granted.

Does that make sense?

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Glad you are joining this discussion @Olivier.Pereira.

I see you concerns with regards to this aspects. I tried to reflect the positions laid down in the expert's statements above with my wording. I think the statements are reasonable since the state of this often globally used closed source is different from the local instance of internet voting software. But I also see it as a somewhat easy escape out of a potentially tricky problem: As you say, it's just not feasible to obtain and publish the source code of any of these standard closed software.

I am not quite sure what could be done to address your concerns, though. Do you think open source software ought to be preferred generally and sub systems with software under a restrictive should always be justified?

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Hi @christian.folini, Thanks for welcoming me despite my late arrival!
My concern is essentially that the wording may be interpreted as an implication from "widely used" to "widely scrutinized". I definitely agree that it is a good strategy to take advantage of the level of scrutiny that external components received independently of their use in a voting system. I think that what matters is the level of evidence that can be obtained that there really is a high level of scrutiny of the chosen components, and I would not consider "widely used" as a sufficient evidence. But I would keep that question orthogonal of the open source vs. closed source question: I believe that there are good and bad examples on both sides.

Christian Folini @christian.folini · 5 months ago Maintainer

Yes, that makes a lot of sense.

How about this wording?

> It is acceptable to use widely used standard components such as operating systems, databases, firewalls, etc. without publishing their source code when there is evidence, that these systems have seen a lot of scrutiny elsewhere and the voting system can profit from these efforts. Technical means to enforce the use of a specific build of said software would be useful.

I'm refraining to use "high level of scrutiny", since we are already asking for evidence. OK?

Feel free to chime in on 6B, 6C and 6D as well. Last Call does not mean that we are done. It means it is a last call for additional arguments or support / criticism of the draft summary.

I would welcome to hear other opinions on the rewording of this paragraph. Was this also a pain spot for other experts that is now mediated? Or would you rather not ask for too much effort in this direction (and focus the effort on areas where the system provider can really make a difference)?

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

I like the proposed rewording, thanks!

Bryan Ford @Bryan.Ford · 5 months ago Developer

Looks good, but I would clarify the last sentence a bit more by changing "a specific build" to "a specific, public and widely-used build".

Christian Folini @christian.folini · 5 months ago Maintainer

Thanks Bryan. I think that makes a lot of sense.

I'm still waiting for more opinions, but if no negative feedback comes in, I plan to adopt the new wording with your enhancement.

Christian Folini @christian.folini · 5 months ago Maintainer

Adopted now.

Bryan Ford @Bryan.Ford mentioned in issue #56 (closed) 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Maybe one comment about this formulation:

"All developed source code and everything needed to understand, to install and to maintain the voting system has to be published. This is meant to allow researchers or auditors to run an election or vote in their own premises with exactly the same reproducible build of the software that is used by the official system."

I think there is another aspect to this, and this is that with public access comes public trust: There will be many computer nerds out there, who will *not* want to build, run, or test the system, they just want to know that the right provisions have been taken, to protect vote privacy and election integrity. It is those people, who will then convince others that the system satisfies certain minimal standards, and hence contribute to building trust among the public.

Christian Folini @christian.folini · 5 months ago Maintainer

Thanks for chiming in @Carsten.Schuermann, but is not this aspect covered with the third paragraph:

> Being as transparent as possible will help to build confidence towards the voting system. Not necessarily because people will actually review the system themselves, but because they know that other people with the necessary know-how and interest have the possibility to do so.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

You are absolutely right @christian.folini. Please disregard my comment.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

No worries @Carsten.Schuermann. It is very helpful you are reviewing all these summaries for arguments that are missing in the summaries.

Christian Folini @christian.folini · 5 months ago Maintainer

I have updated the summary based on input by @Olivier.Pereira and @Bryan.Ford. See above.

Leaving this discussion open for final review.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

"But there are also experts who see this as an information leak that brings an advantage to the attackers, since they are no longer forced to run extensive probes to gather this information."

Maybe it is just me, but while this might be true, it reads a little bit as "some experts support security by obscurity". Could we maybe strengthen or just remove this sentence?

Christian Folini @christian.folini · 5 months ago Maintainer

Good one. I also thought of obscurity when I worded it like this, but I think it is not as simple.

The argument has been mentioned here in this thread and I think it is definitely worth considering, so it should be part of the summary, but maybe in a better wording.

To pile up on this: In one of the reports about the PIT, there is a remark that Swiss Post had to disable certain mechanisms it used to defend against attackers on the perimeter. In fact Swiss Post had implemented a mechanism that would detect probes on the application level and block the IP addresses really fast. This has several interesting effects. One of them is: If there is an unusually large number of IP addresses blocked, then you know that a dedicated attacker with the ability to use rolling IP addresses is probing the system. Information as this would otherwise be lost in a big amount of security alerts due to the continuous probing.

So you place a mine field in front of the attackers. By observing the attackers finding their way across the mine field, you learn more about them. This is definitely more than security by obscurity.

TL&DR: Let's try and strengthen / explain this as you propose:

> But there are also experts who see this as an information leak that brings an advantage to the attackers, since they are no longer forced to run extensive probes to gather this information. Forcing them to probe can reveal useful information about the attackers in return, so this practice should not be seen as a mere "security by obscurity".

@Carsten.Schuermann: Could you live with this?

If I do not hear any negative feedback, I'm going to incorporate this into the summary.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Ah, ok. This was not clear to me. The idea that allowing the adversary controlled access to create a profile of the attacker seems reasonable, but it also comes with some dangers, which directly affect the quality of the election: Blocking IP addresses might disenfranchise certain groups of voters, violating the requirement of universal suffrage. It is easy to imagine these (well-intended) techniques can be weaponized.

Christian Folini @christian.folini · 5 months ago Maintainer

Yes, the method has to be used carefully of course. And I agree that it would be useful to review the method from this angle.

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any additional feedback and given the previous feedback was already in favor of the summary, I am closing this discussion.

Thank you for the intense discussion here and the support to polish the summary.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 6B - Development mode as a means for transparency and public scrutiny (Block 6 - Development and Publication)

Reference to originating discussion block

Block 6 - Development and Publication

Questions

Transparency and public scrutiny are seen as core requirements for an internet voting system and there is a desire to see transparency adopted as a general mindset.

What are the requirements to guarantee continuous interest of qualified security experts in the publicly available code and documentation? (please also see block 7 that touches on this topic)

Does the licensing scheme (closed source vs. open source) influence your answers? What would some kind of open source license change? Which aspects of the system (if any) would an open source license improve?

If we assume an advantage due to an open source license, which parts of the system profit the most from that advantage or where is it really essential to have such a license?

Which level of detail is necessary for researchers to see and understand the development of an internet voting solution? Is there a need to see every commit to the source code, the specifications and the documentation or are consolidated pull requests or tagged versions / minor releases sufficient?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to June 02, 2021 5 months ago

Christian Folini @christian.folini added Block-6 label 5 months ago

Christian Folini @christian.folini mentioned in issue #39 (closed) 5 months ago

David Basin @David.Basin · 5 months ago Developer

I am not an expert on different kinds of licenses but ideal would be a setup where outsiders could download, build, and play with the system. I don't think it is necessary to see the history of commits.

In terms of guaranteeing continuous interest of qualified security experts, I refer to my/Srdjan's response in the original questionare. Namely, this community actually has different subgroups of “security researchers”. The first kind work at universities and are interested in finding security vulnerabilities in complex and important systems and publishing the vulnerabilities they find. Some of these researchers might even use this to showcase their own analysis tools. Most such researchers are responsible and serious and are motivated by publications and also possible positive exposure in the press for their findings. For them, the right to publish their findings is essential although they may be open to accepting a responsible disclosure process. They will largely abide by the law.

The second subgroup is the hacker community who hacks for fame, notoriety, and sometimes for bug bounties. They tend to be less tolerant than the first kind of researcher in terms of restrictions on what they can and cannot do. Members of this community may have an anarchistic streak and even republish the source code anonymously. In general, they have fewer scruples about violating the law. Openness and few, if any, restrictions are important to them. If they are treated well, they may go along with a responsible disclosure process. (So one must manage them!) Both subgroups are helpful in finding problems. The first subgroup may be more successful in finding subtle design errors or bugs concerning the use of sophisticated cryptography. The second kind probably generates more “noise” but also uncovers significant bugs, in particular in the implementation.

Of course, there are no guarantees that these communities will jump in and help out. Sometimes you have to pay for work

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Good you bring back that response, David. I think it makes a lot of sense to distinguish different audiences here.

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

Let me provide some thoughts about open source vs. proprietary software and why open source is not just about the license:

At first sight it seems to be technically irrelevant if you just release the source code of the e-voting system as demanded by the federal law or if you also release the source code under an open source license. But there is an important "added value" provided with the open source license, and that's the community.

Let's start with the definition of "open source". In a narrow sense "open source" only specifies the intellectual property measures of the software license including the rules about releasing the source code involved. Thus it seems like it's just important to follow the strict rules of the OSI (Open Source Initiative, www.opensource.org) approved license and publish source code edits with the binary packages.

But the public source code is just the tip of the iceberg. Although intellectual property is a very important aspect of open source software, today's good practices of the open source development model include much more concepts and practices applied in software development. The community of all contributors is what makes the difference. Thus there are very specific direct benefits of the open source development model:

- *Sharing of knowledge* is very important, but it's not just about releasing the naked source code in one blob as the code review of the e-voting system in 2019 happened. Knowledge sharing in open source communities is also about publishing documentation (API, architecture etc.), providing full access to problem solving processes (errors and their solutions within mailing lists, forums etc.), create ongoing knowledge building activities (tutorials, introduction videos, training videos, published books, blog articles etc.) and so on. And it's also about sharing tacit knowledge, knowledge that includes experience, skills, intuition, creativity etc. This is being done through conferences, workshops, hackathons etc.

- *Collaboration and involvement* within open source communities includes finding common solutions and compromises within a heterogeneous community of backend and frontend developers, security experts, user experience specialists, user interface designers, business process engineers, project leaders, company managers, lawyers, marketing and sales people etc. Thus open source communities usually consists of an interdisciplinary group of people taking care of the holistic, long-term success of the software.

- *Governance and decision making processes* are completely transparent. Thus in a healthy open source community everyone can follow when the core developers discuss about certain strategic issues or code reviewers accept or reject submitted code including feedback provided about the quality and content of the new source code etc. In particular continuous code commits lead to high traceability of the development process and explains why certain code was included or altered. If everyone can trace back the development steps by looking at small diffs in the code then the developer community can understand what algorithms worked and which had to be changed.

These important community benefits of open source projects directly connect with the goals that e-voting actually wants to achieve by just publishing the source code (VEleS article 7a, 7b) according to the introduction of this chapter:

1. gain proper transparency
2. create a higher level of trust
3. grow the number of people who are familiar with the system and online voting technology in general
4. allow authorities to make early improvements in case errors are found
5. achieve continuing valuable public scrutiny

I'm certain this doesn't work by just providing access to the source code and documentation. E.g. if there is a first time Public Intrusion Test (PIT) hyped by the media then professional programmers and experienced security engineers possibly take a look at the code. However, once the source code is completely openly available, after a while the novelty factor and thus the public interest decreases.

There are several examples of important open source projects that have had critical errors for a long time. Do you remember the famous Heartbleed bug within OpenSSL back in 2014? The vulnerable code was inside the cryptography library since 2012. Half a million Internet web servers were seriously affected by this bug. Heartbleed and many more bugs found in open source projects lead to the EU funded "Free and Open Source Software Auditing" (FOSSA) project (https://ec.europa.eu/info/departments/informatics/eu-fossa-2_en) showing the importance of continuous review work and security maintenance of open source software.

Now what does this mean regarding the question about just releasing the source code of e-voting versus releasing the source code below an open source license? Well, if even business critical open source projects may have serious bugs for several years, how

little attractive will a proprietary e-voting system with transparent source code be for professionals to provide "valuable scrutiny"? As it is written in the summary of this chapter "an extensive publication of the source code doesn't guarantee that researchers and third party developers continually participate in the improvement of the code". I'm convinced this is true.

Sure, if money is infinite, why not conducting a continuous bug bounty with large monetary prizes thus incentivising security experts to hunt for vulnerabilities. However, this would at best solve the security issues. It would not lead to a growing number of people "familiar with the system and online voting technology in general" or creating a higher level of trust. Since only a few top specialists would look at the code (assuming they are continuously extrinsically motivated by money...) there would still remain little understanding of e-voting technology in the wider digital community. Thus misinformation and half-truths about how e-voting works would remain since sceptics such as many journalists and hobby hackers would not be informed well.

On the other hand requiring publication of e-voting systems below an open source license could indeed lead to the goals mentioned above. If the community building process is done well, an e-voting system could develop towards an innovative, thriving open source community. Obviously a community doesn't grow just like that. It always consists of different stakeholders with different motives. E.g. civic hackers like the CCC possibly will rather identify code vulnerabilities than program new features.

On the other hand IT companies might look at the code and if it's good enough start contributing improvements since they discover business opportunities. The possibility to provide commercial services for e-voting of the Cantons would incentivise businesses to invest resources into solving security issues. In particular code quality might improve on the long run since companies are interested in efficient customizations and stability for their clients. Knowledge about the source code would spread and grow the community of technically capable e-voting developers.

There are many examples where open source software is being produced by Swiss government. And it's soon being part of the law. Last week the Canton of Bern has initiated the public consultation procedure ("Vernehmlassungsverfahren") regarding the new law on the Digital Administration Act ("Gesetz über die digitale Verwaltung DVG"). Article 24 explicitly covers the aim of releasing open source software by the canton. They also have started a GitHub repository where they publish open source projects (https://github.com/kanton-bern).

To summarize: I'm convinced that e-voting is the best place to require the open source license including its development model since this enables the long-term digital sustainability of the platform.

Edited by Matthias Stürmer 5 months ago

David Basin @David.Basin · 5 months ago Developer

@Matthias.Stuermer points out a number of generally desirable aspects about open source development. I question whether when one wants an open, inclusive community of developers working on a critical system like one for voting.

For example, consider the aspect: "Collaboration and involvement within open source communities includes finding common solutions and compromises within a heterogeneous community of backend and frontend developers, security experts, user experience specialists, user interface designers...". While I can see this as being desirable for many kinds of software, is this appropriate for building critical high-assurance systems where specifications, designs, and proofs go hand-in-hand? How would this work? Are there any examples of this that you can point to?

Collapse replies

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

Thank you @David.Basin for your interesting question! So in CS research there are numerous publications of open source development of security components and business critical software: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=open+source+security+business+critical On the practical side all major security and crypto libraries are developed following open source methods. And from the business critical side the best known software is of course the Linux kernel being used everywhere. Another example is OpenETCS used for controlling trains: "The purpose of the openETCS project is to develop an integrated modelling, development, validation and testing framework for leveraging the cost-efficient and reliable implementation of ETCS. The framework will provide a holistic tool chain across the whole development process of ETCS software, using open standards on all levels." https://itea3.org/project/openetcs.html

So I don't know why the open source development shouldn't work for e-voting systems. Of course Linux nor e-voting is a hobby open source project. Most of today's business-critical open source software is being developed by paid developers, often employed by big corporations such as Google, Microsoft etc. But this is exactly the point: Even the top IT companies develop professional software in the open source model.

Maybe @Eric.Dubuis can explain what you're doing at BFH regarding the open source approach of e-voting?

Edited by Matthias Stürmer 5 months ago

Reto Koenig @Reto.Koenig · 5 months ago Developer

Here, we touch question 6A,B,C at the same posting, as they seem entangled in some way.

@christian.folini : Please feel free to shift it to the most appropriate subtopic.

In our opinion, it is not the running system, that profits most of an open development process, but the risk management behind it and ultimately its project leaders. The earlier such a highly specialized system is developed on protocol-, specification-, and implementation level in an open and highly accessible way, the earlier one (specialists within that field) can intervene in faulty thoughts, weak designs, wrong implementations, and any other shortcoming that lead to an unacceptable solution.

This open process might seem like *publicly performing a naked high-wire act*, but that is simply not true! The shock is much more real ($), if some provider "surprises" the community with a faulty solution in the end, where any correction towards an acceptable solution hints to a complete restart of the project.

One should not be ashamed to be open right from the beginning. Being pointed to some stupid mistake at the beginning of a project is always preferable than to be pointed out (at the very same stupid mistake) in the end, where a complete solution is built upon it.

That being said, e-voting is an extremely specialized topic uniting so many really special skills. Starting with a special cryptographic protocol providing a really perfect specification and implementation skills, where every bit counts up to the usability layer. No single person nor company seems able to cope with all these special skills at once, thus external knowledge and input is absolutely required to master such an endeavour.

Conclusively we advise to be open right from the beginning and do not hinder external experts with legal trapdoors.

Edited by Reto Koenig 5 months ago

Collapse replies

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

I absolutey agree with this statement. Thanks for adding this important topic of continuous transparency and collaboration.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Reto.Koenig .

I see where you are heading, but I am not 100% where you are landing. Can you please try and nail your *open development process* down to one of these options:

- Develop on internal server but publish early, like after every sprint, individual commits not visible
- Develop in public so everybody can follow the development, but code is under a restrictive license
- Develop in public under an open source license

And would that rather be a recommendation or a strict requirement?

Reto Koenig @Reto.Koenig · 5 months ago Developer

> Conclusively we advise to be open right from the beginning and do not hinder external experts with legal trapdoors.

As we are no expert in legal affairs, we *advise* here to go as simple as possible. thus, go for

> Develop in public under an open source license

But in fact, we do not see 'open source license' or 'Develop in public' as a must. If the project owners require it to be more restrictive, then so be it. It might however, scare-off more and more of the experts.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Reto. This is now very clear (and the arguments above all very useful).

(And sorry for being such a nuisance.)

Reto Koenig @Reto.Koenig · 5 months ago Developer

;-) No prob!

Christian Folini @christian.folini · 5 months ago Maintainer

Carsten Schuermann made several comments in 6A that can be applied here as well.

@Carsten.Schuermann : Matthias Stuermer laid down how 3rd participation could be attracted in a comment above. Is this the wishful thinking you have been speaking of in 6A?

Above, I have tried to nail down Reto Koenig and friends (-> BFH) by giving them three options:

- Develop on internal server but publish early, like after every sprint, individual commits not visible
- Develop in public so everybody can follow the development, but code is under a restrictive license
- Develop in public under an open source license

I am reading your statement in way that you would find all three options acceptable with a preference to make individual commits visible. Or do you see a need for a requirement going beyond the first option?

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Thanks @Matthias.Stuermer @David.Basin @Reto.Koenig for an exciting discussion so far.

I completely agree with @Matthias.Stuermer on the importance and benefits of sharing knowledge, of collaborating with and involving the open source communities, and of supporting transparent and open government and decision making processes.

The greatest benefit of the open-source philosophy to software is the transparency it creates. Transparency leads to public confidence and public confidence is the main objective of any election supported by technology or not.

However, I have my doubts that it is possible to create such an open source community around election systems that is trusted to produce excellent code, reliable enough to work towards deadlines, and available. How many cryptographer could implement Beyer-Groth without errors? (I think I could only name a handful, at most!)

Implementing an election system is the pinnacle of programming for the adversarial environment. It is not enough to build something that just works. It is critical, that the system works in all settings, including compromised insiders, cyber attacks, or service outages. Programming for the adversarial environment is extremely hard, and people who can do it properly are rare and in high demand anywhere else.

Some international programming and security experts (for example, many of whom I met at Blackhat/DefCon) do find Internet Voting suspect. In the US, for example, the trend is exactly opposite to Switzerland and away from Internet voting, towards paper ballots, risk-limiting audits, and to disconnect as many systems from the Internet (or local networks) as possible! This means experts, who could contribute to building a secure Internet Voting system, may be hesitant to do so, because it contradicts their inner convictions.

Assuming, that it is possible to attract a small group of expert defensive programmers, the next problem is that of growing dependence on this group: Usually we talk of vendor lock-in, meaning that an EMB grows helplessly depend on a particular vendor and its product, while in this situation, the EMB grows depend on the community to drive the implementation effort along, community lock-in. Just as vendor lock-in, this situation is not desirable.

Lastly, I want to mention the point of governance and open-source projects. I am not sure who this would work for an open-source project. Who is responsible if things go bad? The EMB could encourage the vendor to transition away from a license-fee model toward a more service-oriented business model, asking them to provide an election system made from open-source components. Then it is the vendor, who is responsible for creating (and possibly paying) a open-source community to develop missing open source components, and for delivering a secure and trustworthy product.

In summary, there are plenty of excellent computer scientists, cryptographers, and programmers out there, but I think it will be hard to convince them to join an open source community to implement an Internet Voting system in line with the expectations of the EMB.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Continuous interest: Of course a government can help ensure "continuous interest of qualified security experts" by continuously incentivizing that interest directly, such as by commissioning periodic design or implementation audits, periodic dialogs like this one, and so on. This is extremely valuable and should be done, and adequately budgeted for on a continuing basis, not just as exceptional events responding to crisis situations. For example, expert reviews and dialogs like this - perhaps by a stable but rotating "standing committee" of some kind - might be called on once or twice a year much like an advisory board, and might be invoked for more intensive review and analysis work leading up to significant new design developments or milestones in the system.

Community: I also strongly support the points @Matthias.Stuermer makes about the value of having a community and ecosystem around open-source code, not just the released code itself. Such a community can provide greatly compounding benefits in transparency and security by helping to ensure that many people have their own, independent incentives to understand and scrutinize both the design, documentation, and code on a regular basis for a variety of reasons.

Development process: An open development process is important to this objective - necessary but not sufficient - as already discussed above, in that it enables new people to learn about and integrate into the community gradually, learn how and why

decisions were made by examining the public revision control histories and patch-request discussions, etc. The open development process also gives everyone, including the primary developer/provider of the system, better ability to learn from and avoid past mistakes, by ensuring that the entire history is out the open and making it almost certain that someone will say, "oh, we tried that already, see this thread three years ago, and these are the challenges we encountered..." The larger community can help fill gaps in (though obviously not replace) the "institutional memory" of the design and development process as the small primary development team inevitably turns over with time, for example.

Reusability: As we've discussed in earlier threads especially related to standards, there is great value in transparency and attraction of widespread scrutiny in ensuring that as many critical components in the E-voting system are as modular and reusable as possible in other contexts - such as in other E-voting systems developed and deployed by other governments around the world, by institutional E-voting applications of the type Helios has been widely adopted for, and even by entirely different applications other than E-voting. Of course some parts of the design and implementation will inevitably be extremely specialized, but not all.

Many of the most complex, subtle, and delicate components of an E-voting system, such as the verifiable shuffle and other zero-knowledge proofs in particular, are not inherently specific to E-voting but in principle usable and interesting in many other applications - such as private messaging and group communication, privacy of transactions in digital currencies (which not just public blockchain/cryptocurrency projects but all the major central banks are starting to take serious interest in), and so on. If these critical "foundational" components of an E-voting system can be successfully modularized in an open-source development process and built and maintained in a highly-reusable library separate from and much more generic than the E-voting system itself, then there is much greater chance that many other developers and companies will adopt, join the development in, learn about and regularly scrutinize the designs and implementations of these critical components. This openness- and reusability-focused development will in turn help build the critical mass of support and interest needed to develop formal standards for these components where such standards do not already exist, a (challenging but I think solvable) problem we've already discussed.

License: The openness and permissiveness of the software license is an important enabling factor to community adoption and ecosystem development. Open-source licensing is again necessary but not sufficient, in that a too-restrictive license can easily kill or seriously dampen prospects of independent adoption and involvement, especially by developers in companies who have strong legal concerns about the open source licenses they can and can't accept in software they adopt.

Christian Folini @christian.folini mentioned in issue #41 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Time is running out, so we better get going. Here is a draft summary of this question. I ask you to review it please.

Transparency and public scrutiny are seen as core requirements for an internet voting system and there is a desire to see transparency adopted as a general mindset.

What are the requirements to guarantee continuous interest of qualified security experts in the publicly available code and documentation? (please also see block 7 that touches on this topic)

Does the licensing scheme (closed source vs. open source) influence your answers? What would some kind of open source license change? Which aspects of the system (if any) would an open source license improve?

If we assume an advantage due to an open source license, which parts of the system profit the most from that advantage or where is it really essential to have such a license?

Which level of detail is necessary for researchers to see and understand the development of an internet voting solution? Is there a need to see every commit to the source code, the specifications and the documentation or are consolidated pull requests or tagged versions / minor releases sufficient?

For the experts, it is possible to publish source code with a proprietary license with access to the source code and to receive transparency and public scrutiny in return.

However, an open source system will lead to better results with the goals laid out in the VEleS article 7a, 7b such as transparency, scrutiny, public confidence into the voting system, attracting people and talent to become familiar with the system and voting technology in general. On top, it will also lead to an earlier discovery of bugs, which allows for early adjustments instead of costly repairs relatively late in the development cycle.

A proprietary license with access to the source code can help to reach these goals, but according to the experts, not to the same degree. An open source license is therefore very recommended for an internet voting system.

An open source license would also allow parts of the system to be reused in other open source software, thus making these elements standard components that profit from a wider scrutiny across several projects.

An open source license can also be used to create a community around a system. For one group of experts, this would be very valuable. For the other group of experts, the forming of such a community is not very likely and also not necessarily welcome. These experts are not convinced that a community is able to deliver the quality that is needed for an internet voting system.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also

leave a comment or upvote the summary. If there is or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Replaced "non-permissive license" and "less permissive license" with "proprietary license with access to the source code" after explanation by @Matthias.Stuermer.

Edited by Christian Folini 5 months ago

Collapse replies

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

I agree with @Bryan.Ford that the summary should include a clearer notion towards recommending an open source license for e-voting. I don't see anyone from the experts who recommends not to use open source licenses. Nor I read a technical argument against an open source license.

Regarding the terminology I suggest not to use "non-permissive license" or "less permissive license" when meaning transparent source code (as opposite of an open source license). The word "permissive license" is a well-established term in the open source license area pointing towards MIT, BSD or Apache licenses that allow to include open source software into proprietary software (see https://en.wikipedia.org/wiki/Permissive_software_license). I recommend to use "proprietary license with access to the source code" or something similar if transparency of the code is addressed.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for pointing this out and explaining how "non-permissive" and "less permissive" can lead to misunderstandings.

I'm adopting "proprietary license with access to the source code".

See below for the discussion about the "clearer notion towards recommending an open source license".

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Marius Kobi @Marius.Kobi · 5 months ago Developer

@Matthias.Stuermer describes advantages of an open source software that is licenced under a OSI approved license. In my opinion, most of these advantages only bear fruit if a thriving open source community is established. It is, as @Carsten.Schuermann has pointed out, however, rather doubtful that this will happen. This renders most of the discussion on the advantages of an open source software moot.

What advantages of an open source software (if any) remain if we assume that there will be no thriving open source community?

Which of these advantages cannot be achieved by the publication of all relevant aspects of a system for which no open source licence is granted?

What would some kind of open source license change? Which aspects of the system (if any) would an open source license improve?

In my view, these questions are still open.

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

While I understand the pessimism @Carsten.Schuermann expresses about the likeliness of open source community evolving around an E-voting system, I'm a lot more optimistic on that (and more in agreement with @Matthias.Stuermer on this I think). This is probably an area where we can reasonably "agree to disagree".

For example, even if the specific Swiss E-voting system, exactly as deployed by the cantons, proves to be too specialized in its totality to attract much open-source community interest, nevertheless it may not be so unlikely that (license permitting) some enterprising open-source developer(s) fork the repository and hack it into something more Helios-like to be used by companies and organizations. Even if the latter forked version ends up being substantially different from the E-voting system in use by the government, it will likely also have significant code overlap, especially in some of the most subtle and trust-critical components such as the verifiable shuffle proof code, and thus those overlapping components will get the benefit of wider scrutiny. Thus, even an open source community that evolves not around the E-voting system itself but around heavily-modified forks of it or even just around library components scavenged from it can still provide "community benefits" to the original E-voting system if it is fully open-source.

In terms of concrete suggestions for the summary, I would suggest rewording the first sentence as:

> For the experts, it is possible to publish source code with a non-permissive license and to receive some transparency and public scrutiny in return, although publishing source code with a permissive license will likely receive the maximum transparency and public scrutiny benefits.

Even though the gist of the phrase I added also appears later in your summary, it feels a bit "buried" and worth bringing to the top. In its current form, the first sentence almost sounds like we are *recommending* a non-permissive license over an open-source one, which I don't think anyone is.

Then you can drop the "However," from the next paragraph.

I also have concerns that your phrase "and also not necessarily welcome" may be a bit overly strong, at least based on my reading of the comments above leading to it. Perhaps "and also not necessarily helpful" might be more precise and to-the-point?

In particular, I don't see anyone saying that the existence of an open source community in its own right is likely to be actively detrimental to public scrutiny or transparency. I do see and acknowledge the point @Carsten.Schuermann makes above about "community lock-in" risks. But to me that seems to be about *over-reliance* on (typically unpaid) open source community effort to maintain and advance the development of a software system, which is indeed problematic as we've seen in the case of OpenSSL etc. The existence of an open source community is not the problem here, but rather over-reliance on it to the exclusion of adequate funding or professional development, validation, etc. I don't think any of us would advocate the Swiss government trying to build an E-voting system, release it open-source and wait for an open source community to build around it, and then keep using it but "toss it over the fence" for a unpaid, self-selecting open source volunteer community to maintain exclusively thereafter. That would certainly be foolish and irresponsible in the extreme.

The official, deployed E-voting system must always have the primary support of professional development and validation teams and all the other processes we've been discussing. Provided that remains the case, I do not see any significant risk that the existence of an emergent open source community around the E-voting system would prove actively harmful - but at most arguably perhaps unlikely, and/or unhelpful. We can all reasonably disagree at how likely or unlikely it is for an open source community is to form around an E-voting system, or how helpful or unhelpful such a community would be to the main effort if it does arise.

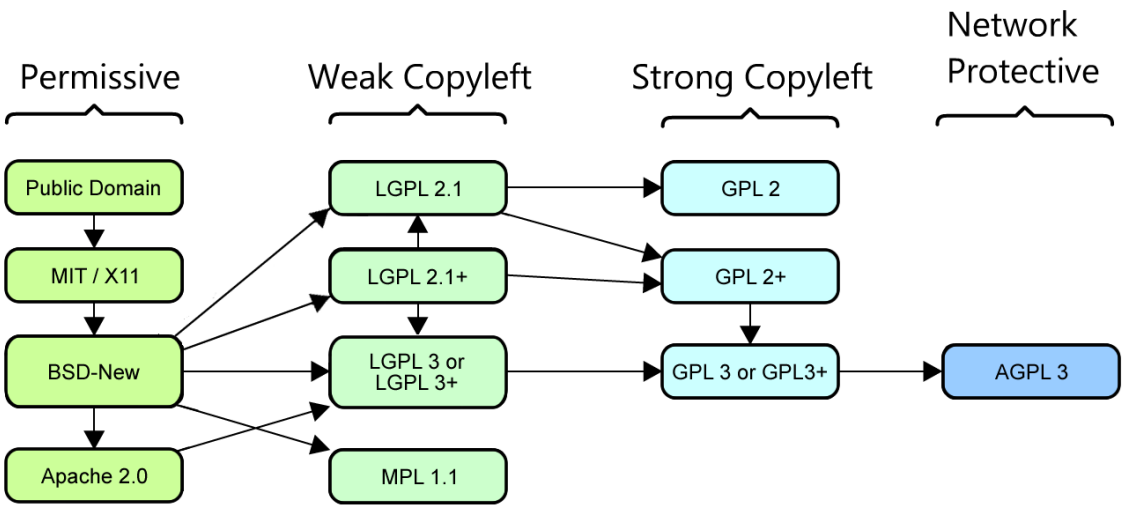
Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

I understand the doubts expressed by @Marius.Kobi regarding the possible lack of an active community around the open source code of the e-voting system. And I agree with him that there is not much added value of making the e-voting system available below an open source license if there doesn't evolve a thriving community around it. Thus it is important to understand some aspects that make industry-specific open source projects succeed in community building:

1. **Generic modules:** As @Bryan.Ford explained very well in his statement above it is important that the released open source components are not too specific but allow reuse und integration in other context. Thus the 'Canton of Thurgau E-Voting Platform' as a whole is not well reusable by other cantons, not to mentions in any other voting context. Therefore there will never by a 'Canton of Thurgau E-Voting Platform' open source community. But if a future open source e-voting system is being released, its core components (which are the most delicate part regarding security and cryptology) could be implemented in other voting systems with high standards. Therefore open source developers could gather around these modules forming together the future open source e-voting community. Modularity and thus reusability are key within the entire open source ecosystem: Research showed that open source software is more modular than proprietary software thus suggesting that generalizability is a success factor for community building.

2. **Key stakeholder commitment:** Like a 'self-fulfilling prophecy' it is essential that key stakeholders really want to build a thriving open source community, especially if the context is rather narrow in the beginning. Industry-driven open source communities need the support of sponsors in order to build an ecosystem of service providers and other contributors. E.g. we investigated the GENIVI Alliance initiated by the car industry. For the success of the GENIVI open source community, it was necessary for large car manufacturers such as BMW, Renault or Nissan to create the governance rules and legal institutions necessary for a professional open source community. There are many more examples in the e-learning or the banking industry where key stakeholders had to express commitment by forming the necessary institutional framework and providing seed funding. Thus the success of an e-voting open source community also depends on the will of the cantons and the Federal Chancellery to really go into the direction of an open source community. Only if a majority of the e-voting responsible persons and agencies are willing to contribute intellectual property (code, documentation etc.), human resources, and funding to build-up and maintain such an e-voting community it will succeed.

3. **License choice:** Industry-specific (or user-driven) open source communities usually consist of different types of stakeholders. Often the users (in our case the cantons) own the software (at least in the beginning) and thus have to decide below what open source license the publish the software components. There are around 70 different OSI-approved open source licenses but only a few of them are used frequently. Although the legal rules are pretty clear and simple there is no 'one-size-fits-all' license model for every open source software. Sometimes it makes sense to choose a permissive open source license in order to facilitate rapid diffusion with the trade-off to receive only little contributions. Sometimes core components are licensed below a weak copyleft license such as the LGPL in order to allow companies to build the libraries into proprietary products. And sometimes strong copyleft or even network protection is required in order to get all contributions back (like the CHVote system) - always bearing in mind the allowed license dependencies:



Source: https://timreview.ca/article/416

The choice of the open source license influences strongly the community intended to be built. It is therefore crucial to carefully consider the possible dynamics that the license (or sometimes several open source licenses) should or should not trigger.

Finally let me address the unanswered questions by @Marius.Kobi:

What advantages of an open source software (if any) remain if we assume that there will be no thriving open source community?
Open source projects without an active community miss the biggest opportunity open source provides. But even with inactive open source projects there is still one major advantage: An open source licensed software permits the user (e.g. the canton) to freely take the code and assign any capable software provider with certain tasks for example to implement and customize the software or fix specific bugs and add features. Thus even if there is no thriving open source community there are still the four software freedoms.

Which of these advantages cannot be achieved by the publication of all relevant aspects of a system for which no open source licence is granted?
If the source code is transparently available but there is no open source license then free use of the software is prohibited, further code enhancements are not allowed, and distribution of the software is prohibited.

What would some kind of open source license change? Which aspects of the system (if any) would an open source license improve?
As explained above the open source license is a necessary but not sufficient condition for building a thriving open source community. If open source is an option it is important to evaluate well the specific license(s) for each of the e-voting components.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for introducing your concerns here on the platform, Marius.

I'll leave it to the experts to address your arguments, as is their role.

If we assume everything hinges on the community, as you say, then I see two ways to address this in the summary:

- Rewriting the recommendation for an open source licence by adding a recommendation to make sure that such a community is in fact being established. This could be linked with the ideas put forward in 6D by @Bryan.Ford and to a lesser extent also by @Carsten.Schuermann.
- Describing the *publication of all relevant aspects of a system for which no open source licence is granted* as equally effective and giving the open source license the status of a special case that only works in special circumstances (-> in combination with a community).

Which one is your preferred option, Marius, and why?

David Basin @David.Basin · 5 months ago Developer

Maybe it is too late in the game to wonder about this... I am in complete agreement with the need for openness (and transparency) but I also appreciate that if a company works to develop software they have a right to protect their IP. So I would be cautious of categorically insisting on a licensing model that sacrifices any kind of IP protection. Is there a licensing model that can satisfy both requirements?

Incidentally, in an ordinance I would not want to insist on a concrete licensing model but rather the properties such a license should have (and have some confidence that the set of license agreements with these properties is non-empty).

Collapse replies

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

Open source licenses are the best way to open up code while preserving IP (for more information about this have a look at e.g https://www.taylorvinters.com/article/an-intellectual-property-law-perspective-on-open-source-software-licences). That's why nowadays all large and small IT companies release software below open source licenses on GitHub and other platforms. I don't recommend to reinvent the wheel by developing a new 'open littlebit' license. Open source licensing is the state-of-the-art way making source code transparent (e.g. SwissCovid mobile app).

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you everybody for chiming in and also those upvoting the original summary.

This is getting a bit complicated and I want to be sure all arguments are being heard and the resulting summary is well-balanced; ideally also speaking with one voice on the core recommendation(s).

As a catchup, here is what happened since posting the draft summary:

- @Marius.Kobi has raised concerns on the discussion leading to the summary and as a result on the summary itself.
- @Bryan.Ford addressed these concerns, suggested to expand the first sentence and replace "not necessarily welcome" with "not necessarily helpful". These rewordings have not yet been adopted, though.
- I have attended a telco on Friday, where Marius signaled consent with the explanations and the proposed rewording by Bryan.
- @Matthias.Stuermer proposed to replace "non-permissive license" with a better wording, which I have meanwhile adopted. See above.
- @Matthias.Stuermer and @Bryan.Ford have both called for an even stronger recommendation towards open source. One that goes further than the original draft summary (still active) or the proposed rewording of Bryan.
- Three people have upvoted the original draft summary: @Bryan.Ford, @David.Basin and @Olivier.Pereira, who did not participate in the discussion here (but is welcome to signal support of course).
- @David.Basin reacted to @Matthias.Stuermer and signaled that he does not want the open source recommendation to be even stronger.

The summary states that an open source license is "very recommended". It expresses a preference, but it does not make it a strict requirement. And reading through the statements above, I think this is as far as we can go until we lose consent.

@Reto.Koenig said it as follows: "But in fact, we do not see 'open source license' or 'Develop in public' as a must. If the project owners require it to be more restrictive, then so be it. It might however, scare-off more and more of the experts." I think this expresses the common ground in this discussion.

So I suggest you let me adopt the proposed rewording(s) by @Bryan.Ford as they seem to have some traction and we do not push it further. Otherwise, we might have to split the position of the experts and I would rather avoid that (even more so as it seems to be working out in almost all of the other discussions).

Christian Folini @christian.folini mentioned in issue #60 (closed) 5 months ago

Marius Kobi @Marius.Kobi · 5 months ago Developer

In my view, the sumamry promotes using an open source license for the internet voting system in a way that is not covered by the comments. I suggest to shorten the summary somewhat and to make it a little bit more balanced.

Suggested wording:

For the experts, it is possible to publish source code with a proprietary license with access to the source code and to receive transparency and public scrutiny in return. According to the experts, a system which is licensed under an open source license may profit from increased transparency, scrutiny and public confidence. This might lead to an earlier discovery of bugs. However, most of the benefits of an open source license hinge on the question whether a thriving community is established. Some experts deem this likely, some are rather doubtful that this will happen and that a community would be able to deliver the quality needed for an internet voting system. A proprietary license with access to the source code can help to reach these goals, but ac-cording to the experts, not to the same degree. An open source license would allow parts of the system to be reused in other open source software, thus making these elements standard components that profit from a wider scrutiny across several projects.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your alternative proposal @Marius.Kobi. I conclude you are no longer in agreement with the explanations by @Bryan.Ford and in support of his extension of the first paragraph.

I see the following changes with your new summary (please correct me if I'm wrong):

- Bryan's proposed extension of the first paragraph that mentions open source is dropped
- The reference to VEleS article 7a, 7b is dropped
- The VEleS goal to attract people and talent is dropped
- Monetary incentives to discover bugs early is dropped
- New: thriving community as a pre-condition to most benefits of open source (-> without a thriving community, open source software is "moot")
- Strong recommendation of open source software is dropped

Here is the `link to a detailed diff`.

@David.Basin, @Carsten.Schuermann, @Bryan.Ford, @Reto.Koenig, @Olivier.Pereira, @Matthias.Stuermer: Do you agree this is a better summary and should we adopt it?

Bryan Ford @Bryan.Ford · 5 months ago Developer

I personally find the original summary, with the minor amendments already discussed, more complete and hence preferable. I don't see any particular need to rewrite it so extensively this late, especially as people's' availability to comment further is likely to be vanishing rapidly.

Matthias Stürmer @Matthias.Stuermer · 5 months ago Developer

As @Bryan.Ford I also prefer the original wording since it reflects better the opinion of the majority of the experts involved in this discussion. People with experience in software development and open source technology are well aware of the opportunities and limitations of community building. The proposed changes would remove important aspects of an open source license and would add a too pessimistic point of view regarding community building.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

I also feel more comfortable with the original summary.

David Basin @David.Basin · 5 months ago Developer

I thought the original summary was balanced and would stick with it.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I also thought that the original summary was good. Let's stick with that.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your confirmations. I conclude the complete rewrite of the summary by @Marius.Kobi did not find your approval.

Meanwhile, I had a very long conversation with Marius on this question here and I have thought about this problem a lot. Several hours to say the least.

I'm sorry, but I am bothering you once more with this.

I'm personally heavily in favour of Open Source. That's why I am paying a lot of attention to avoid anything that could make me look biased in this dialogue and especially here. I have, for example, avoided to smuggle OSS into the summary of the big picture.

Still, Marius reads my summary of this thread here as being biased, since I have avoided to highlight the successful creation of a community as a necessary pre-condition for all the positive effects of open source software. He states that very little in this thread points to the specific case of OSS and E-Voting and that my summary is too apodictic in assuming that an OSS license would bring all these benefits when the experts have linked it with the creation of a community in the discussion.

My reading (and thus my summary) is more like: src code publication is good, OSS is better and OSS + community is even better. But since the successful creation of a community is tricky to demand / require, the community is but playing a secondary role in the summary.

I still think my reading is right and that you support my interpretation. Not the least because you confirmed it.

But maybe it would still make sense to reach out to Marius and reword the 3rd paragraph slightly, being less apodictic and introducing the community idea earlier in the summary.

Original

> However, an open source system will lead to better results with the goals laid out in the VEleS article 7a, 7b such as transparency, scrutiny, public confidence into the voting system, attracting people and talent to become familiar with the system and voting technology in general. On top, it will also lead to an earlier discovery of bugs, which allows for early adjustments instead of costly repairs relatively late in the development cycle.

New proposal

> However, publishing the source code under an open source license would lead to better results with regard to the goals laid out in the VEleS article 7a, 7b such as transparency, scrutiny, public confidence into the voting system, attracting people and talent to become familiar with the system and voting technology in general. On top, it would also lead to an earlier discovery of bugs, which allows for early adjustments and might prevent costly repairs relatively late in the development cycle. At least some of the benefits that an open source license conveys, however, depend on developing a thriving community. Whether such a community will arise, is open to debate.

If we do this, then the amendment to the first paragraph by @Bryan.Ford is definitely off the table since it would be a contradiction to this wording here. But given nobody really supported Bryan's idea(s) above, I was reluctant to introduce it anyways.

Adopting this new proposal above could be a signal that you want to make sure the summary does not suffer from people claiming Christian Folini pushed his Open Source Agenda and this expert advice can be ignored. But you can also state that you this discussion is definitely over for you.

The decision is with you @Carsten.Schuermann, @David.Basin, @Olivier.Pereira, @Matthias.Stuermer, @Bryan.Ford and @Reto.Koenig.

There are six experts active in this thread. If a majority of at least four of you upvotes or responds in favour of an update, I'll adopt it. If I do not hear from you or there is more negative feedback, the proposal is dead.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think my view and reading of the thread is consistent with yours: that public code is good, public code with an open-source license is better, and the latter with a thriving community is still better but not a precondition for open-source licensing being valuable. The vast majority of research software prototypes my lab (and probably most research labs) produces has been released open source but most of it has not by itself resulted in a thriving community, and I expect that is true of most open source software in general. That doesn't mean that the open source licensing is not valuable, because it's also common for an open source software artifact just to get picked up by one or two "downstream" projects or people trying to build directly on it or convert or integrate it into something else, making it more usable or whatever. The fact that the open source licensing makes it *possible* that communities *can* self-organize around any sufficiently interesting software artifact creates the potential for greatly-compounded benefits (and developer effort and scrutiny) if it does happen, but doesn't diminish the value in transparency or benefits only to one or two derivative projects if that's all that happens.

So at least for my part, I'm perfectly happy with the original version of the above paragraph (if you're not going to adopt my suggested edits anyway :)), and don't see a need to weaken it as the "New proposal" seems to.

David Basin @David.Basin · 5 months ago Developer

Both versions are ok with me. But since we are fine tuning, in the sentence: "On top, it would also lead to an earlier discovery of bugs, which allows for early adjustments and might prevent costly repairs relatively late in the development cycle." I suggest changing "would" to "could" since it depends on having a (thriving) community.

Collapse replies

Christian Folini @christian.folini · 4 months ago Maintainer

Well spotted @David.Basin.

Full transparency: I received a proposal by @Marius.Kobi via mail and I adjusted it slightly to give it a better chance with the OSS proponents here in the thread. One such adjustment was to shift from *could* to *would*. @Olivier.Pereira has upvoted your proposal to move back to *could*. Nobody else took it up, so I am going to leave the *would*.

If there was more time, we would continue the conversation. But since we are closing tonight, I need to make a call and I do not see a majority support.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Both versions work for me well.

Olivier Pereira @Olivier.Pereira · 4 months ago Developer

I do not have strong feelings either. Regarding fine tuning, if the new proposal is kept, I wonder if it would not help to be more factual on "Whether such a community will arise, is open to debate." by formulating it as "Whether such a community will arise, is unknown." We can debate probabilities, we can debate strategies that can help the creation of such a community, we just can't know in advance whether the community will arise.

Matthias Stürmer @Matthias.Stuermer · 4 months ago Developer

For me the second version is fine if we find a better wording of the last sentence regarding community creation. From my experience with numerous open source projects in such specific, user/industry-driven context it really depends on the will and funding of the key stakeholders (main contributors, main users such as governments, universities etc.) if a thriving open source community evolves or not.

Thus if the majority of cantons understands the benefits of an open source community and supports conducting such activities, the community will indeed grow and prosper. This requires internal human resources for community management, political and technical leadership, creation and enforcement of governance rules, funding of community activities such as workshops, conferences, hackathons etc.

However, if the majority of cantons is reluctant to embrace the open source development model and does not really support it (with sufficient funding) then obviously no community will be created. It's similar like in other areas such as protection of the environment or traffic safety: Also a community is nothing that just grows like the crop on the fields. A thriving open source community requires active long-term engagement of its stakeholders, especially in a focused area such as e-voting.

Therefore I believe the creation of a community is not "open to debate" or "unknown" if it will arise, but it depends upon the activities driven by the cantons. Thus I recomment to rephrase the last sentence into "Whether such a community will arise largely depends on the community building activities of the key stakeholders" (possibly "key stakeholders" should be named more explicitly "cantons and the federal government").

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you all for chiming in again.

There are six experts in this thread and I called for a clear majority of 4 to have the proposal pass.

Yet I see several neutral positions, a potential vote in favor by @Matthias.Stuermer and a rejection from @Bryan.Ford. Said rejection could be neutralized by Bryan's upvote of Matthias' position, but only if we are sure he means that whole statement and not only the explanations with regards to active community building.

Either way, we are far from a clear majority. Therefore the proposal is rejected.

@Bryan.Ford also proposed a rewording of the first paragraph a few days back. I was holding back to see if anybody would pick it up. Nobody did and I think it would change the weight even further in favor of OSS and we could lose consensus that way. So I think it safe to say that this too is being rejected.

With all this being off the table, I am closing this discussion and I thank everybody for the participation.

Christian Folini @christian.folini closed 4 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 6C - The point in time of the publication (Block 6 - Development and Publication)

Reference to originating discussion block

Block 6 - Development and Publication

Questions

Assuming that neither public development nor an open source license is going to be enforced for the development of the online voting system, the control components and the verifier(s), what is the point in time when the code should be published?

After preliminary internal tests? Before certification or other forms of mandated examination are performed? After a first round of mandated examinations? After all the mandated inspections are over? Before production use?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to June 03, 2021 5 months ago

Christian Folini @christian.folini added Block-6 label 5 months ago

David Basin @David.Basin · 5 months ago Developer

This is difficult to answer without project time scales. The later in the process, the less likely the public is to find problems (which is good). However time may be needed to remediate any problems they find, so there must be an adequate buffer.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree with @David.Basin that this is difficult to answer. As I've stated in 6B, mandating both a fully open-source development process and open-source licenses is by far preferable in general. But in the context of the above question, in particular "*Assuming* that neither public development nor an open source license is going to be enforced" (i.e., *not* a scenario I consider preferable), I would say that the code should be released after a first round of mandated examinations at the very latest. This would at least marginally help reduce the risk of public embarrassment due to the release of seriously-premature code. This public release should also occur before at least one further subsequent mount of mandated examinations (and perhaps more than one if judged necessary), both to confirm and refine the original findings and their fixes and to account for the lessons learned from public scrutiny. But again, it would be much better for the development process and licensing to be fully open from the start.

Edited by Bryan Ford 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for chiming in @Bryan.Ford .

I see a contradiction in your statement above ("should be released after a first round of mandated examinations") vs. your statements in 6B (-> "An open development process is important to this objective - necessary but not sufficient"; "Open-source licensing is again necessary but not sufficient").

Maybe I am just getting this wrong. Are you asking to require open source license, but the master repo is in fact private and 3rd party code contributions are made against a public copy, that has seen a first round of mandated examinations. Or do you have a public repo in mind, but a separate "offical" src code publication site with tagged and examined minor releases?

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I don't have a strong opinion when to release the code in this case (no open/public code). I would require, though, that the code that

was actually used to run the election should be published some time *before* the election. This is not always a given. In Norway, the version of the code published was not the one actually deployed. This prompts two requirements. (1) it is necessary to define mechanisms to attest that the version deployed is also the version published (i.e some form of verifiable attestation). (2) it is necessary to allow for last minute patching of the system without breaking (1).

Bryan Ford @Bryan.Ford · 5 months ago Developer

Thanks Christian. There was technically no contradiction: my paragraph was merely answering the originally-proposed question, predicated on the question's stated assumption that "neither public development nor an open source license is going to be enforced". But I can see how the original wording of my statement could be seen as confusing in that regard, so I expanded and tried to clarify it on that point.

Edited by Bryan Ford 5 months ago

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for the clarification by setting your priorities straight.

Christian Folini @christian.folini · 5 months ago Maintainer

Time is running out, so we better get going. Here is a draft summary of this question. I ask you to review it please.

Assuming that neither public development nor an open source license is going to be enforced for the development of the online voting system, the control components and the verifier(s), what is the point in time when the code should be published?

After preliminary internal tests? Before certification or other forms of mandated examination are performed? After a first round of mandated examinations? After all the mandated inspections are over? Before production use?

With an open source approach, this question would be irrelevant.

If the code is developed in a closed source fashion, then it would be useful to perform a first round of mandated examinations before the publication. That way, less bugs will be discovered by the public and there is still enough time for independent researchers to review the code before it is being used actively.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is or no negative feedback, I will assume consensus and close this discussion.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini mentioned in issue #39 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any feedback, but three upvotes. I am thus assuming consensus and close this discussion.

Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 6D - Assessing the level of public scrutiny and trust (Block 6 - Development and Publication)

Reference to originating discussion block

Block 6 - Development and Publication

Questions

The degree of public scrutiny and the trust the society places into a system is hard to measure. Yet they are important pre-conditions for trustworthy internet voting.

How could public scrutiny and public trust be measured and what do you see as a minimum level?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to June 04, 2021 5 months ago

Christian Folini @christian.folini added Block-6 label 5 months ago

David Basin @David.Basin · 5 months ago Developer

Is it necessary to measure it? What would the units even be?

Suppose the code and all relevant document and artifacts are released and the public is given sufficient time to study them and nobody responds. Provided you are not counting on the public to do your beta-testing and carry out relevant quality assurance activities for you, then no response would presumably be a good thing. In contrast, any real bugs should be remediated.

I strongly doubt they will be no response due to lack of interest: both academic researchers and the hacker community would love to find flaws. If this were a concern, then offer a generous bug bounty.

Christian Folini @christian.folini · 5 months ago Maintainer

Thanks for your take on this David.

It is unclear if there is an absolute need to measure it. But reassurance would be a plus. Even if the units and the minimum level are completely in the dark.

Personally, I have no doubts, there will be interest next year. But I am really unsure if scientists are still interested to study the system in 2026, when all the interesting publications have been written?

David Basin @David.Basin · 5 months ago Developer

@christian.folini : there will still be publications to write in 2026. That is the nice thing about being a researcher in security: our problems are never solved.

Also requirements may change over time. E.g., maybe once e-voting is well supported, it would make sense in Switzerland to study new approaches to elections too given how much we vote here. E.g., random sample elections (where only a subset of the electorate votes), co-voting (where the citizens vote together with their representatives), liquid democracy (where delegation is possible), etc. Economists have good arguments for why such schemes are sensible. Such novel voting schemes give rise to new requirements including new notions of "verifiability", requiring new voting protocols. See, for example, http://people.inf.ethz.ch/basin/pubs/csf18.pdf

Bryan Ford @Bryan.Ford · 5 months ago Developer

I concur with @David.Basin that it is difficult to come up with "formal" metrics of independent public scrutiny - but I can think of many potential *informal* metrics or rough indicators. For example:

- Academic publications, tech reports, or preprints analyzing, or even just citing, the E-voting system and its design documentation over time.
- Journalistic coverage of the E-voting project and its developments and challenges more deeply beyond a pure popular/political level (i.e., beyond "he-said-she-said" clickbait articles).
- In an open-source development process, the number of developers outside of the paid core team who have contributed (e.g., "upstreamed") fixes or improvements to the code, documentation, libraries, etc.
- The amount and breadth of participations in public discussions around such an open-source development process: e.g., the number of people (outside the paid core team) who participate in pull-request discussions, etc.
- The number of developers watching relevant open-source repositories - e.g., "GitHub stars", the number of repository forks or experimental variants appearing in public over time, etc.
- The number of independent companies building and deploying variants of the E-voting system or its components over time, e.g., for institutional use-cases like Helios targets, or other governments adapting it to their own purposes, etc.

I'm sure this is an extremely incomplete and likely "tech-geek biased" list, but it might be a starting point. And of course no single metric or indicator can be taken as reliable or definitive, but only perhaps roughly indicative in combination with others.

Many of these potential metrics or indicators (especially the last) will also take quite a bit of time, likely years, before they become nontrivially useful of course. It takes time to build up deeper public interest and scrutiny.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

@Bryan.Ford mentioned several quantitative measures. There are also qualitative measures, based on interviews, etc. There is an entire field of study dedicated to this called Science Technology Studies (STS). I don't have any concrete recommendation on how to apply the STS methodology here, but it might be a good starting point. I can also dig deeper if necessary. Wiki.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Time is running out, so we better get going. Here is a draft summary of this question. I ask you to review it please.

The degree of public scrutiny and the trust the society places into a system is hard to measure. Yet they are important pre-conditions for trustworthy internet voting.

How could public scrutiny and public trust be measured and what do you see as a minimum level?

There is no ready methodology to observe and measure the public scrutiny that an internet system has received.

However, there are many informal approaches that can help to shape a picture based on quantitative but also qualitative measurements. It would be useful to look into Science Technology Studies to develop a methodology that helps with an assessment of the situation.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is or no negative feedback, I will assume consensus and close this discussion.

Christian Folini @christian.folini mentioned in issue #39 (closed) 5 months ago

Christian Folini @christian.folini mentioned in issue #40 (closed) 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any feedback, but three upvotes from the three experts participating in this discussion. I am thus assuming consensus and close this discussion. Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

7-PIT-bug-bounty.md 18.4 KB

Discussion Block 7 - PIT / Bug Bounty

1. Introduction

The main purpose of the bug bounty program is to motivate public scrutiny of the system. This goal is to obtain a system that is more secure, more transparent and more trusted. It also contributes to building a community of competent people who can scrutinize and evaluate the system.

The importance of scrutiny was already discussed in block 1 as means of making the crypto more difficult to break.

The Public Intrusion Test (PIT) of 2019 was a bounty program which put the spotlight on a specific scope for a limited time. It was not mandated by the VEleS and was run as a pilot project.

The source code was published to satisfy the requirements for transparency and scrutiny of the VEleS but was not in the scope of the 2019 PIT.

Mandated intrusion tests carried out by specialized companies are a complement. They are explicitly required by VEleS for Internet facing systems (scope 5, protection against infiltration) and implicitly for the back-end systems in scope 3 (ISO 27001 audit of infrastructure).

There are very different scopes that can be tested. One is made up of all standard infrastructure that supports the voting application: firewalls, web servers, databases, logging servers, DNS and so on. Another one is the tailor made software that implements the protocol: client side software in the browser, software on the control components, the voting server, the printing office, and others. Finally, documentation, physical security or social engineering can also be inspected or tested.

There are important differences between the scopes. For example, it only makes sense to test the supporting infrastructure that is actually provided by the voting service provider, as the security mainly depends on the configuration of the infrastructure. The security of the software or documentation can however be tested in virtual systems run by the testers themselves, as long as they run the same software with the same software configuration.

While it is possible to run a public intrusion test against the Internet facing infrastructure of the voting service provider it seems unrealistic to expect the provider to open up its backend infrastructure to the greater public for testing.

There are more scopes that cannot easily be tested by the public. Imagine the chaos if the public was invited to test denial of service attacks on the actual infrastructure of the service provider. The same holds for social engineering where real employees or voters would be attacked by the public. It also seems difficult to let the public attack the physical security of the system.

2. Proposition of scopes and types of tests

Based on the responses of the questionnaire and the above statements, it seems that the following would be a good way to organize tests:

- The time and scope limited PITs would become a continuous bug bounty program with a broader scope. Most tests could be run on local copies of the system. This implies that running a local copy of the system should be facilitated by the provider, e.g. by providing preconfigured containers or virtual machines. The testers could then for example also test the security of back-end elements like the control components. The internet facing infrastructure would be tested on a testing infrastructure set up in the same way as the productive infrastructure, although with lower requirements for availability. The specification of the system, the code or the documentation of processes would also be part of the scope of the bug bounty.
- The back-end infrastructure of the provider, denial of service attacks, social engineering or physical intrusion would not be in the scope of the public bug bounty program. They are part of the mandated private intrusion tests.

3. Responsibilities

In the responses of the questionnaire several experts saw the federal administration or an independent entity running the bug bounty program.

Bug bounties are typically run by the provider of the tested systems, for practical reasons.

This contradiction could be solved the following way:

- The federal administration defines the modalities of the bug bounty program (scope, terms and conditions, bounties, …) with the service provider and the Cantons.
- The provider operates the program. This includes assessing the reported bugs, paying the bounties and other operational tasks.
- The federal chancellery assesses the operation of the bug bounty program on a regular basis.

4. Related Questions

The related questions are labelled The related questions are labelled Block-7.

4.1 Individual links to related questions

- Block 7 Discussion A - Evolution of the PIT
- Block 7 Discussion B - Private penetration tests and infrastructure
- Block 7 Discussion C - Delaying the publication of the bugs
- Block 7 Discussion D - Other ways of public participation

5. Questionnaire

This block is based on the answers to questions 4.1, 4.5, 4.6 and 4.7.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
4.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.5	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.6	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
4.7	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

6. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 7A - Evolution of the PIT (Block 7 - PIT / Bug Bounty)

Reference to originating discussion block

Block 7 - PIT / Bug Bounty

Questions

Do you agree that the 2019-PIT should evolve into an ongoing bug bounty program with a larger scope ?

The scope would include everything that can be tested continuously by the public without creating inconveniences for the service provider (for example the back-end infrastructure). This would typically include Internet facing infrastructure reserved for testing, the software, the specification of the software and all security relevant documents (specification of the system, processes).

Do you see value in including a test infrastructure (reverse proxies, firewalls, web servers, ...) set up identically to the production infrastructure into the scope of the bug bounty program?

Since the purpose of this test infrastructure is to test the front-end security, the back-end could not be identical to production. For example, it could not have the same level of redundancy. Would this reduce the value of the tests?

Should the value of the bounties be related to other bug bounty programs (up to tens or hundreds of thousands of dollars) or to the price paid for exploits on the public market (up to hundred thousands or millions of dollars)?

Do you agree that the bug bounty program should be operated by the service provider ?

Do you agree that the federal administration should be in charge of defining the goals and modalities of the program and of assessing the program?

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to July 01, 2021 5 months ago

Christian Folini @christian.folini added Block-7 label 5 months ago

Christian Folini @christian.folini changed the description 5 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

The test infrastructure should be as close as possible to the actual deployment infrastructure in order for the intrusion tests to deliver real value. Any discrepancies may lead to false positives or false negatives in the tests (revealing bugs that won't be there in the real setting, or missing bugs that arise in deployment). I think it would be a mistake to assume that it does not matter if the back-end differs from that used in deployment. I don't have a strong opinion on who runs the bounty program or how much should be paid out.

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

in the order of the questions asked

- yes, there should be a continuous bug bounty program for the whole evoting. the scope is to be chosen according to the possibilities
- yes, there is a huge added value, if the infrastructure within the scope of the bug bounty is the same as the one used for evoting. this should be identical if possible to prevent false positives and false negatives
- if the bug bounty infrastructure (whether front- or backend) does not match the physical infrastructure, the tests should still be performed in a similar environment, as vulnerabilities may be found in both infrastructure. it may be possible to find a format

that allows the back-end systems to be tested as close to reality as possible.

- the bounties to be paid should be motivating and appropriate for the use and the "value" of the evoting. i would base this on common bugbounties of other companies or comparable systems, i do not find it appropriate to base it on the prices of exploits
- in my opinion, the bug bounty should be set up professionally in order to appeal to competent and interested people. it doesn't matter if you use an existing platform, a neutral partner or the service provider. the service provider might be assumed to hide results or not pay out bounties. therefore i would rather tend to use an existing platform or an independent party.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your detailed response @Tobias.Ellenberger.

The maximum bounty is likely to be a heated question. Am I right when I read something in the 10K-100K CHF range as a reasonable maximum bounty? And if correct, could you provide some reasons why it's this number and not 1M?

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

yes @christian.folini this is a hot topic.

i would refer to common bounties and not limit them to an amount.

An amount in the millions or according to the question "how it is paid for exploits" I think is inappropriate. Although it can be argued that for a root exploit on a common operating system a single-digit million amount is paid, and this corresponds to an amount that is paid for larger political campaigns, this amount can - if manipulation of the election is set as a goal - be used more efficiently and effectively (e.g. companies for data analysis).

Furthermore, these are only the "out-of-pocket" costs for the exploit. The successful (undetected) orchestration and execution of an attack, in which the exploit itself is an important but small part, are much higher. Therefore, in my opinion, there is no reason to pay out such high amounts for bounties.

The question of who would finance such a high bounty is left out.

Edited by Tobias Ellenberger 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your arguments. This is helpful.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

The bug bounty program should be a long term and attractive program. Money is certainly an important incentive, but not the only one. I agree with the Tobias' amounts and arguments. We might also consider additional forms of rewards, like an official recognition from the Swiss government for what has been done: e.g., appearing on an official board of recognized contributors, receiving a digital signed document stating the contribution, etc. This does not replace the money reward, but can give extra incentive as it might be used later by the white hat to improve his/her own reputation (for example to be hired by a specialized company in the future).

Barbara Erni @barbara.erni · 5 months ago Developer

Thank you for your statements @Oscar.Nierstrasz and @Tobias.Ellenberger. I have a follow-up question: If the test infrastructure has to be more or less identical to production (including back-end), then costs might be rather high. A permanently available test infrastructure also requires a certain level of support. During voting phases, priority has to be set on the productive infrastructure (not to be confused by the public and voters with the test infrastructure). How would you assess the cost-benefit-ratio of a permanent bug bounty program for the internet facing infrastructure (evolution of the 2019-PIT testing a system identical to production during one month into a permanent public test of a production-like test-infrastructure) considering that the published code will allow conducting a complete ballot (c.f. discussion block 6 and question 6A)? Do you see alternatives to a permanent bug bounty for the internet-facing infrastructure (mandated tests, hackathon, etc.)?

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

The key point for me is to be prepared for major changes in the evoting system itself, or of the infrastructure it is built upon. A successful evoting system will continue to be under development, especially in a country like Switzerland, where votes are carried out multiple times per year. As the system grows and changes, new threats will arise, and measures will need to be in place to allow

aggressive testing on a regular basis. I cannot judge the cost-benefit of having a test system in place to enable bug bounties four times a year (one possible extreme) vs once every two years, but this is something that will have to be not only considered but also reviewed regularly.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I concur with @Oscar.Nierstrasz and @Tobias.Ellenberger that it is highly valuable to have a test infrastructure matching the production infrastructure as closely as possible, ideally available continuously or at least regularly.

The necessity to prioritize support of the production infrastructure during voting phases might reasonably justify dialing back the support of the test infrastructure - e.g., temporarily disabling it, reducing its capacity, or leaving it in operation but in an explicitly low- or deferred-support state (i.e., "if it breaks no one is going to fix it or answer questions until after the election").

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

@barbara.erni : thank you for your questions:

> During voting phases, priority has to be set on the productive infrastructure (not to be confused by the public and voters with the test infrastructure).

during the election the "PIT environment" should not be available in my opinion, among other things for the reason you mentioned. However, it should be ensured that the infrastructure used for the election has been tested in advance in the same version (e.g. same software versions).

> Do you see alternatives to a permanent bug bounty for the internet-facing infrastructure (mandated tests, hackathon, etc.)?

over time it will become apparent which is the "best" (including cost-benefit ratio) setup for the tests. A bug bounty and mandated testing are complementary in my opinion, as is the disclosure of code. In addition, a hackathon or similar can be done for "special cases", if none of the already mentioned methods is suitable. This can be e.g. the testing of single backend components or a possible closed source. The advantage of a permanent bug bounty (with the always latest version) is that - if the community is active - new vulnerabilities can be tested relatively quickly and thus detected and implemented. It may also be possible to go as far as checking which attacks are carried out in which way in order to learn from them and take them into account in future development. In contrary to a "short" test period, this also avoids that attacks which are more complex cannot be carried out (due to time limitations) and that people do not have time to participate during this period.

[EDIT by Christian Folini]: Made sure quoted questions and response can be distinguished.

Edited by Christian Folini 5 months ago

Reto Koenig @Reto.Koenig · 5 months ago Developer

We will here provide our view for 7a,b,c,d as our answer covers all four topics alike.

@christian.folini : Please feel free to rip it apart.

In our opinion, the only purpose of the PIT is to answer the following question: Can the service provider cover the required trust assumptions made by the e-voting protocol.

Unfortunately though, this can only be done by disproof, hence, by demonstration that reality is too weak in one or the other point to withstand.

As already pointed out by the other experts within that track, this urges the PIT to be permanent available / accessible, without any restrictions what so ever. Thus, a system must be in place acting as the "attackable twin". To remain practical, however, it seems reasonable to provide a user deployable system being ready-to-run at a click at any user's side. This way, even hefty social engineering attacks can be simulated without endangering any operator at the provider's side. Here, the term 'user' is assigned to anybody in this world who is interested be it a single person or an entire organization.

In order to be able to simulate the organizational units and their tasks at the user's side, the system provider is required to provide full and detailed documentation of the used hard- and software, as well as the organization units, their abilities, tasks, duties and their standard operational procedures (SOPs). This documentation must be made public, as it reflects the organizational knowledge of the internal attacker.

Please note, we do not see the PIT at the user's side to be equivalent in quality as the PIT at the provider's side. But we propose some procedures in place, where attacks can be simulated on the 'own' running instance first. If a 'successful' attack-path is found, this attack can be sketched by the attacker and sent (confidentially) to the provider. If the provider does not react within a reasonable time frame or does not accept the attack-path, the attacker can publish the attack immediately and make it available for public scrutiny. If the provider accepts the attack-sketch, the attacker is allowed to execute the attack on the provider's "attackable

twin". If successful there, the quality of the attack can be considered as very high and requires immediate action by the provider. This allows the service provider to cope with the 'patch-gap'. But, no attack-path is allowed to remain confidential for ever, and eventually must be disclosed after a reasonable time.

This way, the quality of an attack-path is becoming measurable. Not only in the damage caused by it, but also in the time it has to remain confidential. This, of course could be mapped to a monetary base.

Edited by Reto Koenig 5 months ago

Collapse replies

Rico Mazzoleni @Rico.Mazzoleni · 5 months ago Developer

Thank you for your input, @Reto.Koenig .

The system provider's infrastructure contains many security elements (WAF, firewalls, monitoring, ….) that could not be part of the ready-to-run user deployable system. Therefore it is hardly possible to build up the exact same infrastructure and organization (despite the documentation being published) in the user's environment.

Is our understanding correct that the "attackable-twin" would be used to avoid any examination on the user's environment and that the user has to prove on the "attackable-twin" his attack works (and not the other way round as the system provider doesn't know the user's specific environment)?

How do you assess the risk of a user claiming publicly he could hack the system (actually in his environment that does not include all the security measures taken by the system provider) without checking on the "attackable-twin" system? And how can this possibility be adressed in advance?

David Basin @David.Basin · 5 months ago Developer

Some care must be taken in setting up intrusion tests and evaluating the results.
For the proofs, one assumes that various components are corrupted and the security rationale does not involve firewalls, monitors, etc. So attacks found on the system without firewalls, monitoring, etc. are certainly meaningful. Moreover, as @Vanessa.Teague pointed out, attackers need to be given the ability to corrupt all the untrusted components. Nevertheless, practically, for defense-in-depth, it is also important to test the infrastructure in as realistic a configuration as possible. I imagine this could be done by distributing the system and infrastructure in a virtualized, networked environment.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think it is important to distinguish the "core" from the "peripheral" infrastructure. (Anyone, feel free to suggestion better terminology.)

To me, **core** infrastructure is everything the E-voting system components that must be (collectively) trusted for election integrity and voter privacy (but not for availability), as well the underlying trusted hardware/software stacks those components directly depend on. The core infrastructure is what must be operating correctly (at least to an adequate threshold) in order for the assumptions of the E-voting system's security and privacy proofs to hold.

The **peripheral** infrastructure in contrast is everything required to keep the core infrastructure running smoothly from an availability perspective but is formally untrusted with respect to the E-voting system's security and privacy proofs. This includes examples like network routers, WAF, firewalls, network monitoring, performance analysis and debugging tools, etc.: basically everything trusted for availability (keeping power, network packets, etc., flowing smoothly to and from the core infrastructure) but that should not need to be trusted at all for the election's integrity or voter privacy (a complete compromise of the perimeter infrastructure should at worst halt the election, and never compromise election integrity or voter privacy).

I think that for the *core* infrastructure, the suggestion by @Reto.Koenig of having an identical or near-identical "attackable twin" available most of the time (at least outside of election periods) is a realistic and extremely valuable goal.

But I think @Rico.Mazzoleni is right that including all the *peripheral* infrastructure would probably be too costly and unrealistic, and could decrease the infrastructure operators' ability to maintain dynamic information advantages over potential attackers (i.e., attempt to ensure that attackers must perform risky and potentially-detectable probing and do not just "know" everything that is running and how it is configured). So I think it is reasonable and appropriate to exclude the peripheral infrastructure from the scope of the "attackable twin" objective. Further, the "attackable twin" should probably even be topologically located in a fairly different and separate network environment from the main production system - perhaps in a DMZ-like environment with deliberately lowered perimeter protections, perhaps in a public or private cloud environment to facilitate efficient high-bandwidth DoS attack scenario tests locally without risk of impacting production infrastructure or the provider's Internet connectivity, etc.

So in summary, I think that provided a clear and appropriate distinction is drawn between "core" and "peripheral" infrastructure, the core can and should definitely have an "attackable twin", while the peripheral infrastructure need not and probably should not.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I agree with many of the things said above, esp. @Tobias.Ellenberger. Here I just highlight the following:

Generally, the more you allow for scrutiny of your actual system, the more one seems to benefit from a public intrusion test.

> Do you agree that the bug bounty program should be operated by the service provider?

Primarily, the bug bounty has to be run professionally, with contributors being compensated for their work. I would advise to partnering up with a well-respected bug bounty provider. They will not only be able to facilitate insight into the best way to organise this but will also have a community of bug bounty participants that will scrutinize the systems. It may also be possible to learn from other public administration agencies that have previously done bug bounties, such as the US Department of Defense (Hack the Pentagon).

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

@Florian.Egloff: Just to be sure: "Partnering with a well-respected bug bounty provider" means that weaknesses with a Swiss online voting system are first disclosed to a foreign entity. Correct?

(Even if the provider does not do the triage, it is easy to prepare the BBP platform in a way the foreign partner see findings before Swiss authorities do.)

Florian Egloff @Florian.Egloff · 5 months ago Developer

Honestly, I am not attuned enough to know whether there is a Swiss bug-bounty provider. I recognize the trade-off you are alluding to. perhaps others have more experienced with running bug-bounties and whom to best partner up with?

I agree with @Tobias.Ellenberger that the service provider is probably not the "right" party to run it (for conflict of interest reasons). There may be other parties competent enough to run it, perhaps with an established bug-bounty provider consulting on how to best do this? All I wanted to highlight is that this involves more than making it accessible to the public: you want eyeballs actually hitting your test, and that is perhaps easiest when you can tap into an existing community. Whether this necessitates having an established provider running it or not, I do not know.

Christian Folini @christian.folini · 5 months ago Maintainer

Thanks Florian. I just wanted to make sure this is not left without the context.

For the record: Bug Bounties are something new in Switzerland. Very few programs, very few discussions around it.

There is no Swiss Bug Bounty provider in sight. French yeswehack.com is making some progress in Switzerland and I hear that the big 2 American players are looking to enter the market as well.

The PIT in 2019 ran on an adhoc platform set up by SCRT SA under the supervision of @Sergio.Alves.Domingues. So SCRT would do the triage and report to FedCh directly.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

If the system claims to defend against certain kinds of insider attacks, for example claims verifiability against an attacker who controls parts of the server-side infrastructure, then the PIT should provide the 'attackers' who are examining it with the same access as the attacker the system claims to defend against.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

@Vanessa.Teague: Does "same access" necessarily mean access (physical access!) to the production server or can it be a copy of the production server or would a container with identical binaries be equally adequate?

Bryan Ford @Bryan.Ford · 5 months ago Developer

This is where an "attackable twin" located in or directly connected to a cloud environment as I suggested above could be particularly useful. Public researchers would then be able to run high-compute-load, high-bandwidth attack simulations directly against the "attackable twin" without risk of impacting security or availability of infrastructure outside this cloud environment.

One example of an academic cloud/testbed environment specifically designed to facilitate "attack simulations" like this is ISI's

DeterLab, a facility my lab at EPFL has been using for many years in our research. This might be used as a model or inspiration for such an environment might look like, although a testbed infrastructure specialized to an E-voting system of course need not have all the generality and functionality of a testbed like DeterLab.

Sergio Alves Domingues @Sergio.Alves.Domingues · 5 months ago Developer

I agree with most of the statements made by @Reto.Koenig above regarding the availability of a continuous Bug Bounty program as well as the delivery of user-deployable instances of a similar setup.

Vulnerability research is commonly performed on local instances of the target software and having the capability to deploy and access all the components of the system is thus required for this research to be most efficient. The same goes for the documentation and procedures. Indeed, as stated by @Vanessa.Teague, if insider attacks are to be considered by the tests, it is mandatory to provide the testers with meaningful access levels. Moreover, keeping any documentation or procedure "secret" would suggest that part of the security relies on this secret.

Regarding the subject of who should operate the bug bounty itself, my feelings are not so strong. Relying on well-known BB platforms has some advantages (e.g. outsourcing the time-consuming initial triage) and may certainly help in attracting "quality" researchers to the program. This attraction will however be very dependent on the bounty values. On the other hand, as mentioned by @christian.folini these platforms are not based in Switzerland which could be a drawback (if only in terms of trust).

On the other hand I don't see as a major issue if the bug bounty is operated by the provider itself, as long as the researchers are inherently free to disclose their results after a (reasonable and pre-established) period of time. Indeed, even if the provider did not acknowledge some vulnerability, the researchers could still make it public and demonstrate their results.

Whoever the "operator" of the bug bounty itself, it seems however mandatory to me that the goals and modalities are set directly by the Federal Administration and that the latest closely oversees it's operation.

Bryan Ford @Bryan.Ford · 5 months ago Developer

On the bug bounty topic, I think that appropriate amounts should be evaluated not just as a single "moment-in-time" decision, but dynamically and adaptively in the context of the longer-term development of the E-voting system, and particularly based on its maturity.

One key reason that full kernel exploits for major operating systems are paying on the order of ~1M bounties is because these kernels have received many years of hardening and maturing under intense long-term scrutiny. This maturity is reflected both in terms of the code itself, and in terms of the internal development disciplines, peer-review, testing, and signoff processes of the companies maintaining these kernels. Because of this maturity, new full exploits almost never consist of just one single "eureka" discovery anymore, but instead must be painstakingly constructed from several (often five, six seven, ...) hairline fractures, each of which might individually seem almost harmless, but only when extremely cleverly combined actually produces an exploit. E-voting systems are, let's face it, years away from this level of maturity and hardening.

For this reason, I agree with opinions above that lower bounties are probably appropriate now - but the explicit plan and "roadmap" should be for those bounties to increase as rapidly as feasible with the maturity of the system. For example, maybe the bounty should be only ~10K in the next year or two, but should be raised to ~100K in the next 2-3 years, and perhaps should set a target of reaching the ~1M range in say 3-5 years once the design and implementations are highly mature, multiple diverse and ideally formally-verified implementations exist, etc. In other words, the bug bounties will necessarily have to start at a moderate level but there should be a "graceful escalation" plan to increase them.

On a related topic, @christian.folini is it appropriate for a discussion of potential approaches to financing bug bounties to occur somewhere? Of course since this dialog is among a group of technical rather than finance experts anything we might suggest cannot be taken as definitive. But nevertheless there are important interactions between the technical side and the finance side that seem relevant as I've brought up earlier: e.g., exploiting technical diversity to increase the effective protection provided by moderate-size bug bounties using techniques like those used in the Hydra framework.

Using these techniques could help address both the real and perceived risks of discovered bugs being "first disclosed to a foreign entity", as well as the real or perceived risks of having the bug bounty provider potentially (perhaps necessarily) not itself being a Swiss company. If by the construction of the E-voting system and its bug bounty program it should be nearly-infeasible for any single discovered bug to compromise the E-voting system as a whole, then the potential incentive for either a foreign hacker or an insider within a bug bounty provider to exploit a discovered bug secretly, rather than using responsible disclosure to claim the bounty, could be reduced by orders of magnitude.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

@Bryan.Ford: We are aware of your analysis of this problem and your proposal in the questionnaire.

The summary of the questionnaire for 4.7 sums this up as follows:

> One expert states that the bounties must be much larger. He also proposes an insurance model that could keep the budget relatively low but still guarantee large payouts (“Hydra framework”), thus setting the incentives in a way that all findings are reported.

The dialogue only has room for so and so many questions discussed at large. So we had to drop certain items and this was one of them. However, given it is part of the summary it is definitely not forgotten.

Are you OK with that for now?

Bryan Ford @Bryan.Ford · 5 months ago Developer

Sounds reasonable, thanks Christian.

Bryan Ford @Bryan.Ford mentioned in issue #35 (closed) 5 months ago

Bryan Ford @Bryan.Ford mentioned in issue #37 (closed) 5 months ago

Bryan Ford @Bryan.Ford mentioned in issue #52 (closed) 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Here is my view on PITs and bug bounty programs.

General points:

First, I want to point out how important it is that there are clear rules of how to disclose vulnerabilities and how those who disclose are protected, something like a Security Vulnerability Disclosure Program. These rules should hold independent of any PIT and any bug bounty program, and should include a responsible disclosure policy. Before all else, those who disclose need to feel secure. No one wants to be prosecuted and arrested for their work.

Second, I believe that those participating in a PIT subscribe to a "hacker's code of conduct" of ethical hacking. For example, if you (as a hacker) find a vulnerability, you don't break the system, you let other find it as well. When you install third party tools on the target to help your attack, you tidy up after you are done, etc. There is no need to tell the community what they are allowed to do and what not (as it was done during the 2019 PIT). Hackers might not listen, and I am sure that they are very capable of organizing themselves.

Third, a PIT or pen test is a snapshot in time. The conclusions that one can draw from such a test, are limited to the attackers knowledge, how up to date the hacking tools are, and how clever the pentester is. The result of a PIT may be invalidated tomorrow when a new vulnerability is released, and it doesn't say anything about zero-days in circulation.

Specific comments:

I agree with several comments above, that a PIT should be ongoing, and that the target should be an "attackable twin" of the production system. However, I do see the PIT primarily as a tool for showing the presence of problems rather than their absence. Therefore a PIT works well for shutting a system down (if it has bugs), but it does not work well for creating confidence in the system (at least for me). There are better ways of doing this as we have already discussed in an earlier discussion point.

One thought I had is that the public interest in participating in a PIT may be reduced, if there is no clear start and end date.

Regarding bounty programs, I think they are good to have, but I have no comments about how to organize them. I also don't have a strong opinion about monetary rewards attached with bug bounty programs. Food for thought: The EFF homepage Security Vulnerability Disclosure Program, for example, asks the community to understand that they don't have any money to pay out (as they are a not for profit organization). I think it is also worth considering if the bug bounty budget could be used more effectively in other ways (i.e. towards building more public confidence), for example, for verifying critical parts of the code base, generating mix-net implementations from formalization in proof assistants or similar. One million euros can go a long way. Lastly, I believe that the bug bounty program should be operated by those stakeholders, explicitly mandated in the Security Vulnerability Disclosure Program.

In summary, I believe, most importantly, that the federal administration together with the Cantons should define such as Security Vulnerability Disclosure Program that holds for all of Switzerland. This Program should include provisions for PITs and the bug bounty program.

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

Regarding Internet facing infrastructure:

There seems to be some disagreement in regards with including the "peripheral" infrastructure in the scope of a continuous public

bug bounty program.

I would not go as far as Bryan and say that it is reasonable to exclude peripheral infrastructure as it is not part of the trusted elements.

One reason why we can pretend that some elements are trusted is because of all the layers of peripheral infrastructure that makes it virtually impossible to hack into a control component and steal a private key.

Last week a vulnerability was discovered in Palo Alto firewalls that allows an attacker (in some situations) to gain control of the firewall (CVE-2020-2021). If the only thing protecting the control components from the Internet where five layers of Palo Alto firewalls then we would have to question the trust assumptions.

There is also a public relation value in letting the bounty hunters try to penetrate the Internet facing infrastructure. It is the most visible attack surface and many people believe that hacking a crypto protocol starts by breaking through the firewall.

In my opinion, there would be value in having the identical twin fitted with the identical Internet facing infrastructure. Bounty hunters should be offered a bounty if they can bypass the firewall/WAF/etc, even if this does not allow to break verifiability per se. The infrastructure would not need to be available 24/7/365 for being valuable and the details would have to be worked out with the provider.

Could we agree that there is a need for the Internet facing peripheral infrastructure to be part of the scope of the bounty program even if not available all the time ?

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

I certainly agree with you that it is *desirable* "for the Internet facing peripheral infrastructure to be part of the scope of the bounty program even if not available all the time". I just feel that there may be reasonable arguments that it may be unrealistic to *mandate* that *all* of the Internet-facing peripheral infrastructure to be in-scope. To try to clarify this, the position I would propose is:

- It is *mandatory* that *all* of the trust-critical core infrastructure be in scope of the bug bounty program (and ideally available most if not all the time via a "hackable twin" for example).
- It is *desirable* that *as much as feasible* of the peripheral infrastructure and its configuration also be in scope, while accounting for limitations due to cost or other considerations.

Does this make sense and seem reasonable?

Philippe Oechslin @philippe.oechslin · 5 months ago Maintainer

Regarding the back-end infrastructure:

Vanessa makes the very valid point that if we claim to protect against insider threats we should give this access to people who want to verify that.

While I think that it is mandatory that the back-end it audited through appointed pentests, I understand that the providers will not agree to have an open house event where bounty hunters from all over can roam through their back-end. I also do not believe that it is feasible to create an identical twin of the back-end because of all peripheral services like monitoring, logging hihg-availability etc.

A solution would be to ask the provider to document a set of controls that are in place in the operational back-end (eg. segmentation, firewalling, encryption). The mandated pentest would include a validation of these security controls. People would try the hack their own copy of the back-end. They would be rewarded if they find an attack that is not blocked by the documented security controls.

Christian Folini @christian.folini · 5 months ago Maintainer

Let me try and wrap up this discussion. I am not 100% sure I am writing a coherent summary here and I am also not sure, you have formed a consistent opinion. The dilemma is the nature and use of the possible bug bounty targets: an attackable twin and / or a containerized version that everybody can download and play with.

From my perspective, I think it would be a reasonable position to say that one or the other would be enough. However, reading through all the responses I see many statements that make the attackable copy a necessity. But there is an equally long list of statement that explain while a local copy for the attackers is welcome, but it seems not capable of replacing the attackable twin.

My interpretation is thus, that you as experts see a need to have both.

If you think, that the idea of the attackable twin is much too expensive, the return-on-investment will be too small and that the local

copy is absolutely good enough, then please say so and we can rewrite the summary.

Do you agree that the 2019-PIT should evolve into an ongoing bug bounty program with a larger scope ?

The scope would include everything that can be tested continuously by the public without creating inconveniences for the service provider (for example the back-end infrastructure). This would typically include Internet facing infrastructure reserved for testing, the software, the specification of the software and all security relevant documents (specification of the system, processes).

Do you see value in including a test infrastructure (reverse proxies, firewalls, web servers, …) set up identically to the production infrastructure into the scope of the bug bounty program?

Since the purpose of this test infrastructure is to test the front-end security, the back-end could not be identical to production. For example, it could not have the same level of redundancy. Would this reduce the value of the tests?

Should the value of the bounties be related to other bug bounty programs (up to tens or hundreds of thousands of dollars) or to the price paid for exploits on the public market (up to hundred thousands or millions of dollars)?

Do you agree that the bug bounty program should be operated by the service provider?

Do you agree that the federal administration should be in charge of defining the goals and modalities of the program and of assessing the program?

The publication of the code, mandated testing and a Bug Bounty Program are all seen as complementary. An ongoing Bug Bounty Program replacing the former Public Intrusion Test is seen as a welcome evolution.

The tests executed by the independent bounty hunters are useful to check the trust assumptions of the system and they support a defense in depth even in a world where everything is formally proven.

The target of the attack tests should not be the system used for the voting itself. Instead, the security researchers should have the options to attack a separate copy of the electronic voting system as well as a deployable version of the system that they download and run on their own premises.

The electronic voting system is divided in core elements and peripheral sub-systems. An attackable copy of the production system has to consist of the core elements and all the necessary peripheral sub-systems. It seems reasonable to concentrate on the voting here and skip auxiliary functionality (e.g. high availability). But there is also a danger in an attackable copy with a simpler setup, as this could lead to false claims of a successful attack against the bug bounty system. Such claims could be hard to refute.

There might also be reasons to pause the operation of this attackable copy during active voting / election periods. Alternatively, the support level of this target systen could be reduced during these periods.

There are going to be limits as to which access is granted on the attackable system in order to test potential weaknesses against insider or social engineering attacks. Providing a ready to deploy version of the voting system can be a useful alternative that allows to test for these weaknesses. It has to be accompanied by a complete documentation of the used hard- and software as well as organisation units, their abilities, tasks, duties and standard operational procedures. This would allow a researcher to design and test an attack on his or her own premises while still shaping it according to the operation of the production voting system. Again, false claims of successful attacks against this local copy could be hard to refute.

The disclosure policy for the vulnerabilities should be covered by a policy that also encompasses other vulnerabilities of the system. This policy has to be defined by the Federal Administration.

There is no clear opinion whether the system provider should run the Bug Bounty Program itself or a third party partner; possibly commissioned by the Federal Administration. However, it is very important that the Federal Administration defines the goals and modalities and oversees the program.

The size of the bounties should be adaptive and grow with the maturity of the system.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation either in writing or a simple upvote. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT] Typo

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

I have not seen any feedback, but two upvotes on the summary. The summary has been posted five days ago, but since this is a very important question and there have been many participants, I leave it open for the time being and invite you to either comment or confirm you are OK with the summary.

Christian Folini @christian.folini · 5 months ago Maintainer

There are 5 upvotes now and still no negative feedback after a full week. I'm calling it a day and close this discussion.

Thank you for participating and your support.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 7B - Private penetration tests and infrastructure (Block 7 - PIT / Bug Bounty)

Reference to originating discussion block

Block 7 - PIT / Bug Bounty

Question

Do you agree that independently of the bug bounty program (PIT), the Internet facing and internal infrastructure should be subjected to private penetration tests regularly and at each substantial modification? These tests would include controlled DDOS attempts.

Note hat in block 6 we have a discussion about the publication of the reports of these tests.

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to July 02, 2020 5 months ago

Christian Folini @christian.folini added Block-7 label 5 months ago

Christian Folini @christian.folini changed due date to July 02, 2021 5 months ago

Christian Folini @christian.folini changed the description 5 months ago

Sergio Alves Domingues @Sergio.Alves.Domingues · 5 months ago Developer

Conducting penetration tests against the infrastructure (both Internet facing and internal) should indeed be a mandatory requirement. These tests should occur not only after substantial design or configuration changes but also at regular intervals even in the absence of any modification (to account for the discovery of new vulnerabilities and attack techniques).

This is a common practice for most companies and there is, in my opinion, no obvious reason why such a critical system should not at the minimum be submitted to this type of tests.

Regarding "controlled" DDoS attacks, I understand that this refers to typical flooding-based attacks. In that case, the interest of simulating these attacks seems less obvious to me. Indeed, it only makes sense if a specific anti-DDoS solution or service is used and is meant to be tested by the attack. However, in that case the simulation must be capable of matching the order of magnitude of an actual attack for the test to really be relevant.

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

yes agreed. the tests should be carried out regularly (continuously?) Furthermore, weak-point findings should be checked after they have been remedied. The intervals should be chosen appropriately. It is important that several parties are taken into account to avoid "blind spots".

The operator or a responsible person should check new attack possibilities, which become known, asap for the applicability of the infrastructure.

A difference should be made between "standard checks (penetration tests using a given method for the purpose of comparability) and concrete attack attempts (ethical hacking) in a controlled manner = the broadest possible scope, but with as few restrictions as possible (e.g. omitting social engineering).

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your contributions guys. You both have a pen-testing background, so I take it with a grain of salt that you think this is a good idea :)

Are you interested in responding to the follow up question: It is unlikely that a provider can allow unplanned DDoS attacks against its infrastructure or its Anti-DDoS provider (possibly even for legal reasons). The same is true with social engineering campaigns executed by bounty hunters. In both cases - and possibly some additional ones - you can only test this via a mandated contractor.

Now do you see it viable to publish this report afterwards? Would this have any weight? Would you as pen-testing company agree to have your reports published? Would that make them more expensive?

And lastly, a bit a diabolical question: Do you think your reports would carry any weight with a wider audience?

Collapse replies

Sergio Alves Domingues @Sergio.Alves.Domingues · 5 months ago Developer

Making the reports public seems appropriate and I don't see any obvious reason why this could be a significant problem for the contractor (at least it would not be for us).

Regarding social engineering attacks, care should just be taken to perfectly anonymize any published report and prevent the identification of the actual targeted persons (especially in case of successful attacks).

Florian Egloff @Florian.Egloff · 5 months ago Developer

As advised above, I would also advise hiring competent penetration testing firms to assess the security of the system in addition to a PIT. The two operate complementarily to one another.

AND: Yes, the findings should be as transparent as possible, whilst taking care to not endanger currently live systems. Thus, the feedback has to go to the maintainer of the system, the risk owners (election officials of the cantons), and as a matter of course, to the certifier of the system. Ultimately, the findings should ultimately be made publicly available, including responses by the system maintainers/operators on how issues were addressed.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I concur with all the main points made above, but don't have anything significant to add since pen-testing is not my expertise area.

The infeasibility of allowing uncontrolled DDoS or other attacks against production infrastructure in part motivates my suggestion in 7A of cloud/testbed support for more open, unscheduled, and un-mandated attack simulations against an "attackable twin".

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Bryan, I see the advantage of the cloud-twin you described in 7A. Yet, if we talk about DDoS, then the difference of the cloud infrastructure make the results very hard to use for the production system:

- If there is a finding, the service provider might blame it on the cloud.
- If there is no finding, then this will help to foster a false sense of security.

This difference is also there with other aspects of the peripheral systems, but it is really striking with DDoS, I think.

Bryan Ford @Bryan.Ford · 5 months ago Developer

The point is that a cloud environment makes it possible to simulate potentially large-scale DDoS attacked safely, demonstrate proof-of-concept effects and amplification factors, and answer "what-if" questions like, "how big of a global DDoS botnet would a foreign adversary likely need to have in order to make the E-voting system inaccessible via this particular form of DDoS attack?" Any time you do a system it's always necessarily up for debate whether the simulation's scenario assumptions are "realistic". But a well-designed simulation experiment does not just pick an arbitrary point in this space, rather explores many potential points along various axes: e.g., presents graphs relating the effectiveness of potential attacks against the strength of the simulated adversary (botnet etc), and then lets the reader or policymaker decide which points along that graph are realistic enough to worry about.

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

Each security audit method has its specificities, advantages and limitations. Mixing different approaches (e.g. bug bounty and pentesting) definitely helps getting an accurate picture of a given environment's security posture.

Christian Folini @christian.folini · 5 months ago Maintainer

Time to wrap this up. Here is my draft summary.

Do you agree that independently of the bug bounty program (PIT), the Internet facing and internal infrastructure should be subjected to private penetration tests regularly and at each substantial modification? These tests would include controlled DDOS attempts.

Running Penetration Tests on a regular base is seen as standard industry practice that should be applied here as well. It adds a targeted method to the mix and complements a continous Bug Bounty Program in order to get a more accurate picture of the security posture of a system.

If done right, penetration tests also allow to examine aspects of the system defenses that can hardly be tested in a bug bounty program (e.g. DDoS, social engineering, etc.).

All the reports as well as the responses of the system provider should be made public.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation either in writing or a simple upvote. In case there is no feedback or no negative one,I will sooner or later assume consensus and close this discussion.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any feedback, but two upvotes. So I am assuming consensus and close this discussion. Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 7C - Delaying the publication of the bugs (Block 7 - PIT / Bug Bounty)

Reference to originating discussion block

Block 7 - PIT / Bug Bounty

Question

Do you agree that the disclosure of a bug could be delayed by a limited time (e.g 90 days), while it is being assessed or while there is an ongoing votation that is vulnerable to the bug?

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to July 03, 2021 5 months ago

Christian Folini @christian.folini added Block-7 label 5 months ago

Christian Folini @christian.folini changed the description 5 months ago

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I don't have a clear opinion on the ideal time frame, but I think it's definitely important to define a clear procedure to communicate regarding possible problems. Taking into account that information may be leaked to the media.

Christian Folini @christian.folini changed title from **Discussion 7C - Evolution of the PIT (Block 7 - PIT / Bug Bounty)** to **Discussion 7C - Delaying the publication of the bugs (Block 7 - PIT / Bug Bounty)** 5 months ago

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

yes. a vulnerability disclosure process should be defined and published. The time to disclosure should be chosen appropriately, taking into account the severity of the vulnerability and the complexity of the remediation.

Christian Folini @christian.folini · 5 months ago Maintainer

We're easily talking of large numbers here:

```
90 (evolving industry standard)
30 (voting period)
20 (~ setup)
30 (~ legal appeal period)
-------------------------
~ 1/2 year

(It's probably an exaggerated example, but I would like to test the reasonable limit.)
```

Is this acceptable from your perspective, or will a system provider have to cut this down?

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

I think this is really complicated, and much more so for voting systems than for ordinary software. Remember that most of the traditions around the responsible disclosure process relate to ordinary commercial software, and that the primary tradeoff is between A) the risk that the vulnerability will become known to bad people, who exploit it, vs B) the risk that the vulnerability will remain hidden from ordinary users, who are endangered by it.

The perfect unachievable ideal would be to tell all the potential victims immediately, so they could defend themselves or take evasive action, while hiding it from all potential attackers indefinitely, or at least until it was patched. Obviously this isn't possible, so we make some approximate heuristics, but it's important to understand that they're only approximate, that these two different objectives might tilt in very different directions in particular circumstances, and that election software is a very unusual circumstance.

You probably know that several people at SwissPost and Scytl were annoyed that Sarah, Olivier and I went public quickly with our findings. You may not know (but it's true) that we got a fair bit of equally-firm feedback in the opposite direction from the security community, who questioned why we had delayed at all. I'm not (still) trying to win the argument – I just want to lay out why there are arguments on both sides, and what sorts of considerations should be relevant, with last year's findings as a running example.

First observation:
If the system is not running, and not expected to run until all observed vulnerabilities are patched, then neither (A) nor (B) applies.

If you think about it, there's really no good reason for any delay in the setting we *thought we* were dealing with – one in which the vulnerabilities didn't affect any running systems. There's also no particular reason for haste.

Second observation:
You don't really know how many other systems are affected by the same problem, so assessing (A) and (B) is hard.

I was probably more astonished than anyone when I learned that New South Wales was already running a system affected by the same bug. Verifiable mixing really doesn't make much sense in the context of their protocol, but they'd bought it anyway and not told anybody (which makes even less sense). Nevertheless, after our first finding went public, they announced that they were affected too.

I would have gone public at the same time even if I had known, because in this case I think the important thing was to tell the public that there was a problem, but I can immediately see that there might be disagreement. If nobody in New South Wales had heard about the problem until after a patch was in place, that might arguably have reduced the risk that a corrupt official or external attacker exploited the problem, but it would also have increased the risk that the problem went unnoticed all through a real election – we would have been relying on Scytl to convey the information to other affected parties (specifically, the New South Wales Electoral Commission) and given that it was their failure I'm not convinced of the justification for assuming that they would do it properly.

Third observation:
If you know that another system is affected, and is already running, then you have a real dilemma

When we found the second problem, which related to the Zero Knowledge Proofs for decryption, we knew that it affected the SwissPost-Scytl system (which was not in use) and suspected it affected the New South Wales system, but didn't know for sure. So we notified both authorities simultaneously. We then went public straight after the New South Wales election closed. This is because (A) ceased to be relevant, but (B) was supremely important, because it's exactly the sort of failure that a disappointed candidate might use in a challenge to the election result. (In this case, nobody did.)

It's also worth noting that the ZKP problems are insider attacks, so telling only the insiders doesn't guarantee objective (A) either – you might be telling exactly the people most likely to exploit the problem. So in the case of insider attacks the dilemma is even harder.

[btw, SwissPost dealt honestly with the problem, while the New South Wales Electoral Commission put out a press release declaring that it was "not relevant" to them. This could not be disproven until months after the election, when they did finally make their code available and it could be immediately shown to be subject to the vulnerability. This story alone is justification for making all election source code openly available, and I hope it illustrates why I don't always assume that the election authorities are perfectly trustworthy for the purposes of accurate disclosure.]

Fourth observation:
If the problem affects past systems not currently in use, then there is really no reason for delay

The last discovery (which Swisspost figured out at about the same time we did) was that the ZKP problem affected the client-side proofs in a system that had already been used in Switzerland, but was not in use at that time. Again, given that (A) doesn't apply, (B) favours immediate public notification.

So in summary I would say:
In general, 45 days is a reasonable rule of thumb, but there are many reasons to vary this in particular circumstances, for example:

1. If the software has not yet run, there is no reason for either delay or haste. (If you're not sure whether the software is running elsewhere it might be reasonable to take a few days to try to find out.) Neither (A) nor (B) applies.

2. If the software has already run an election, but it has finished, then there is a strong obligation to disclose promptly, within the

time that a disappointed candidate could potentially use the information to contest the result. Objective (A) is no longer relevant, so (B) becomes the overriding concern.

3. If the attack is an externally-exploitable vulnerability, and the software is running, then one argument would say that the right thing to do is notify the election authorities and let them decide what to do about it, but another defensible position would be that it was better to tell the public and recommend that they don't use the system. Both (A) and (B) apply, and the right balance isn't clear.

4. If the attack is only an insider attack (for example, an opportunity for a mixer to forge a proof), and the software is running, then I think it seems wrong to tell only the insiders and not the public. The usual rules of responsible disclosure don't really cover this case. Objective (B) is obviously extremely important, and objective (A) is not helped by secrecy. In the case of the New South Wales decryption proof bug, we chose to wait until the election was closed, then immediately make it public. Again the right balance is unclear.

Edited by Vanessa Teague 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Wow, that's an impressive coverage of the topic and it is interesting to see which arguments you take / took into consideration.

Just a small followup question: You are mentioning 45 days a reasonable rule of thumb. The question mentioned 90 days; a number based on an evolving industry standard. The number has also been mentioned in the responses to the questionnaire.

So how do you arrive at 45 days as a base and would 90 be equally reasonable or is that too long given the e-voting context?

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

I actually hadn't noticed the 90 days in the question, sorry - I'd picked 45 days because that's what was in the PIT terms (and also the iVote terms). I don't think it matters - the point is not so much the default as the understanding that there are lots of circumstances in which it doesn't apply.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you very much Vanessa. I thought it might have been the PIT / Src Code program.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Great analysis @Vanessa.Teague and I don't have that much to add.

One further dimension that maybe almost in "goes without saying" territory, but is probably worth bringing out explicitly, is deciding *how much* to reveal publicly at which particular time. This seems complex as well.

For example, should the *existence* of a (believed, or confirmed) discovery be publicly disclosed earlier (perhaps much earlier) than the technical details are released? On the upside, disclosure of the existence of a vulnerability may help add public pressure to the provider and/or government bodies involved to deal with the issue promptly one way or another, thereby perhaps somewhat addressing the question of whether insiders to the process are really trustworthy in this respect. Also on the upside, allowing a substantial delay between announcement of existence and announcement of details might help address the long time-scales that @christian.folini mentions above.

On the downside, public disclosure of a vague report without details could undermine public trust in the election system (especially if an election is currently in progress), could fuel speculation and conspiracy theories, etc. Further, even just an existence announcement might arguably trigger hackers and even potential nation-state adversaries around the world to "smell blood" and go on an exerted hunt, even based on only the vaguest information about the issue.

I really don't know where the right balance is here. Maybe @Vanessa.Teague or others could comment on that as well.

Collapse replies

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

@Bryan.Ford I agree this is a conundrum when the issue has not been fixed, and I don't have a clear answer overall. One more point on the upside: people who have multiple voting options might choose to use a different one, even if they don't know exactly what the problem with the electronic channel is.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I'd like to add one comment to @Vanessa.Teague's excellent analysis. Every Internet-Voting system relies on trust assumptions that can often be paraphrased as "vulnerabilities that are accepted and considered non-exploitable by those who commissioned the system".

Some of these trust assumptions might be foolish, for example, when assuming that just because the vendor stores votes in an Oracle database they are secure and accurate, or that there is no insider threat. Others can be more reasonable, such as the public ID infrastructure is good enough to guarantee voter eligibility.

Clearly the foremost objective Internet Voting requirements engineering should be to hold the set of trust assumptions small, sane, and reasonable. A conundrum arises, however, when a newly identified vulnerability was previously deemed acceptable: To disclose and discredit the EMB, or not to disclose against one's own convictions.

Edited by Carsten Schuermann 5 months ago

Collapse replies

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

@Carsten.Schuermann I'm not sure I understand the situation you're describing here - I'd assume that the trust assumptions were public and explicit (that's been a theme of many of the other channels in this discourse). So I don't really see how a non-public trust assumption could suddenly become invalid - all the trust assumptions should be public, right?

It's up to the citizens, not to those who commissioned the system, to decide what vulnerabilities and trust assumptions are accepted (and, as @David.Basin correctly points out, there's a duality between modelling a likely attacker and delineating acceptable trust assumptions).

But in any case, I don't think 'not embarrassing the authorities' should be a goal of responsible disclosure. (Not endangering the ordinary users is the goal.) And I think that scientists should avoid, as much as possible, entering into contractual agreements that introduce this conundrum. The rule should be that the public knows the whole truth about their voting system. (Subject to (A) above.)

David Basin @David.Basin · 5 months ago Developer

Another way to look at trust assumptions: they are dual to adversary models. An adversary model states what we believe the adversary is capable of doing.
A trust assumption states things we believe (or assume) the adversary is incapable of doing. Hence just as it is clear we want to design systems that are secure against the strongest possible adversary, they should be secure with the weakest possible trust assumptions. Practical security, e.g., penetration testing can refute our trust assumptions by showing that real adversaries had some capabilities we had not attributed to them (e.g., extracting keys via side channels). This is one of the advantages of testing, even in a world where everything is formally proven (WRT a model of the adversary).

(Maybe this post doesn't go here, but it is a supplement to @Carsten.Schuermann's post.)

Christian Folini @christian.folini · 5 months ago Maintainer

Time to wrap this up. Here is my draft summary.

@Vanessa.Teague has given a very concise overview covering many, many factors that influence a flexible response to the initial question of this discussion. I have not repeated or summed up these reflections. Instead I name the different factors. I think that is adequate for a summary, since the complete text will be made available to the decision makers as well.

Do you agree that the disclosure of a bug could be delayed by a limited time (e.g 90 days), while it is being assessed or while there is an ongoing votation that is vulnerable to the bug?

It is important to have a well defined and transparent disclosure process. The process and the individual decision about a publication should be based on industry best practices.

Yet the high relevance of the electronic voting system and the clocked production use of the system make it a special case.

There are two conflicting desires that have to be balanced:

- Potential attackers should not get information about an active before it has been fixed.
- The public at large should get the information about a vulnerability so it can react accordinly.

Depending on the point in time, previous or scheduled use of the system (domestically or abroad), the severity of the vulnerability and the complexity of the remediation, the response should be different.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation either in writing or a simple upvote. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT] Wording of the two conflicting desires based on the proposal by @Bryan.Ford below.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

"two contradiction desires" -> "two conflicting desires" or "two opposing desires"

Christian Folini @christian.folini · 5 months ago Maintainer

Oopsie. Thank you for pointing this out. I have adopted "two conflicting desires".

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Just one question @christian.folini. What do you mean by "the public can react accordingly"? What can the public do?

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Sure. This picks up on @Vanessa.Teague stating: "potential victims immediately, so they could defend themselves or take evasive action"

I have used a more general, but perhaps not ideal wording, since the public can also discuss it once it becomes known.

What do you think?

Christian Folini @christian.folini · 5 months ago Maintainer

Time is flowing quickly and I am wrapping up discussions across the platform. I think we're done here and I thank you for your contributions.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 7D - Other ways of public participation (Block 7 - PIT / Bug Bounty)

Reference to originating discussion block

Block 7 - PIT / Bug Bounty

Question

Do you see other ways of motivating the general public to learn about and improve the security of the system?

Here are a few examples to start the discussion:

- Run a hackathon to develop voting clients that facilitate testing of the servers
- Run a hackathon to write the main parts of the crypto code of the control components in a different language
- Run a contest for improving the current solution (e.g. for a more secure printing office)
- Create a contest for social engineering plots and reward the best ones according to some defined criteria (e.g most impact for least number of people fooled).

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to July 04, 2021 5 months ago

Christian Folini @christian.folini added Block-7 label 5 months ago

Christian Folini @christian.folini changed the description 5 months ago

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

As I've already said at several places, I think that a citizen science project might be a promising way to involve the general public, and not just a very specific group as is the case in hackathons (though I agree hackathons are a good idea). I hasten to say that I've never done a citizen science project myself, so I might be too optimistic. But I do think they could be an appropriate instrument to involve the general public in a structured way, under the guidance of experts.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

All the proposed actions would involve the *general* public only indirectly via the media. Is the goal to raise awareness and stoke interest? In that case it would be important to talk with the media to try to understand what kinds of initiatives would interest Urs and Ursula Schweizer. I don't think the general public cares about hackathons or citizen science projects, or even really understands what they are about. I am personally at a loss as to how to excite the general public about IT outside of the scope of action movies and TV shows.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I agree with the general sentiment of this comment. The reason why I think a citizen science project might have potential is that it would involve people who are more representative of the general public than a hackathon, even though of course they would still be strongly self-selected. And it could generate the kind of media attention that -- I agree -- it's needed for the general public to know or care about the issue.

But the idea of a citizen science project is not just about raising awareness and generating media attention. It's also about getting normal people to use the public board in a structured setting, under the guidance of experts.

Tobias Ellenberger @Tobias.Ellenberger · 5 months ago Developer

@christian.folini what is the meaning of "general public"? is it the Swiss population or the it security scene or interested professionals?

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

I think it is in the spirit of the question to read *general* in a very general way. So it could be anyone within Switzerland and beyond. @Fabrizio.Gilardi is thus right on target with the idea of a citizen science project.

However, if you look at it from the angle who actually has the capability to participate and contribute to the technology itself, then I think a certain focus on security people and interested professionals makes sense as well. That's also how the examples were chosen.

It's an open question (I admit not all questions in the dialog are so open ●) and broad responses are welcome.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I think it would be worthwhile to think how the interested voter can become part of the verification ceremonies, just like "Stimmenzähler" are elected on a communal level to "verify" the result.

[I didn't see where to best place the suggestion, so I posted it here.]

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

Just to make this even clearer: I would advocate for transparency and voter inclusion in every stage. Note that the function of a “stimmenzähler” and the possibility of a “wahlbeobachter” is not to have the most secure version of a process, but to have the most democratic accountability of a process. The same principle should, in my opinion, apply to electronic voting procedures.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

Agree. There should be a structured way for interested voters to participate.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I will expand even more here (This could equally belong to #8).

Many of the questions asked seem to adopt the mindset, which assumes, that improving certification, using more secure cryptography, and getting more publicly verifiable notice boards will lead to more trust in the system. That may be part of the answer. However, a socio-technical view of voting would stress other factors, such as the social context, including political structures that have enabled trust in the current voting process, despite its (demonstrable) insecurities. Thus, under such considerations, other topics would be of relevance, including what political work is necessary to earn voters trust? How does the e-voting technology need to be designed for and by the Swiss political process? How are voters part of that design process? How is the system shaped by the localities, including the village/cantonal level politics?

@christian.folini: where will we be discussing such issues of process? There is still no clarity of whether there is a plan to earn the voter’s trust and what that looks like. The new channel can be objectively the most secure channel, if there is no social trust in it, then an attacker has many opportunities to discredit it. (this touches on issues raised by @Fabrizio.Gilardi) elsewhere.

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

possibly this could be relevant here too: #22 (comment 988)

Copied in here for reference:

> As stated elsewhere (#37 (comment 949)), I would couple the need for usability with the need for a clear strategy of how trust in the system can be formed. This goes beyond the "technical" analysis of whether the system is trustworthy, but has to include a political process for desigining&introducing such a technology.

> One idea would be to work with communal councils (Gemeinderäte). Gemeindeversammlungen are great places to reach the interested voting public. One could start by building touchable systems: i.e. replicate the system with physical objects for a workshop-like experience. Give people the option to “touch” the system logic. Explain how trust in the system is established & give people the option to “practice” internet voting right there.

Florian Egloff @Florian.Egloff mentioned in issue #22 (closed) 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

To add to the list of potential ideas in the original question:

- Make available some publicly-accessible cloud/testbed infrastructure supporting safe and efficient attack simulations against an "attackable twin" of the real system - perhaps using that in hackathons, contests, or even security/crypto courses at universities to enable hackers and students everywhere to learn about the system and explore new potential attacks.

The discussion above about how to get the *general* "non-geek" public involved in some way is challenging but also extremely important. But here are a couple of ideas:

- Organize a "sponsorship" process in which the family/friends/neighbors/communities of the individuals or teams involved in a hackathon can sign up to support the team, perhaps just by publicly showing support as a fan or supporter, perhaps by making a small donation toward the individual's or team's attendance of the hackathon if travel is required, etc. If organized well, this could give the tech-savvy participants both reason and incentive to reach out to their local communities (i.e., seeking sponsors), in the process having to explain to and help educate their communities about what E-voting is, what the security issues are at a high level, what they're trying to achieve in the hackathon, etc. In the bigger picture, mechanisms like this could help build the "social trust network" needed to build trust in the system beyond those with the technical knowledge to understand it directly.

- Another potentially-meaningful way to get the general public involved is to convene, perhaps periodically, a citizens' assembly to study and comment on the E-voting system and program from a "general public" perspective. Such a citizens' assembly might take the form of a moderated, deliberative dialog not unlike this one (although perhaps using a more user-friendly and less developer-oriented framework like Loomio instead of GitLab). The key difference would be that the citizens' assembly would ideally be composed of randomly-sampled "ordinary citizens" of a suitable population (e.g., one or a set of participating cantons) rather than domain experts. Over the course of a few weeks or a couple months, the citizens' assembly members would first learn more deeply about the background, technology, and issues around the E-voting system, assisted by presentations and discussions with both insider and independent experts, lawyers, politicians, and others with different perspectives, and would collectively work to synthesize some form of "citizens' perspective" on the system or program. Not any kind of binding up-or-down decision, of course, but more of a report that reflects the range of the general public's perspectives and concerns with some confidence of proper statistical representation. Even if only a small sample of the general public actually participates in such a process, the broader public trust can still benefit from (a) the perception that the general public's opinions have been adequately considered and taken into account, and (b) even a small random sample population can greatly help "seed" productive communication and education throughout their neighborhoods (after they go home and report at their universities and town halls what they learned and how it went, etc.) I know a number of academics in political science and related disciplines who I'm sure would be happy to help design and run processes like this.

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I endorse these ideas.

Florian Egloff @Florian.Egloff mentioned in issue #24 (closed) 5 months ago

Bryan Ford @Bryan.Ford mentioned in issue #50 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I have my doubts that public participation in improving and securing the system is possible or desired. It assumes that the public wants to do it and that there is general public support for the project. I do not know about the support for Internet Voting in

Switzerland, but I wouldn't be surprised if there are people who are vehemently and outspokenly critical against it. I am afraid that whenever an event is organized, such as a hackathon, those critical toward Internet Voting will attend and use it as a platform to further their own cause.

What would be the point of involving the public in securing the system? To create trust? I doubt that this is possible. Security does not necessarily imply trust. Otherwise elections that are secured by military, for example, in many post-conflict and developing countries, should be trusted much more than they are. I also want to mention that security experts understand that zero-days, known to be in possession of some powerful Nations States, can be used to attack any kind of system. Hardening a system against honest but curious insiders is one thing, securing it against a full-blown Nation State attack another.

The best way to engage with the public is therefore to explain what evidence an Internet Voting system generates, why this evidence is sufficient to claim that the result is trustworthy, how this evidence can be checked, and invite the public to develop their own third-party checkers. Then of course it must be possible to run those third-party checkers on live data after an election, but this discussion is best left for another discussion point.

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

It's certainly true that there are people "vehemently and outspokenly critical" of E-voting, and basic principles of democracy demand that those voices be heard and taken into account. However, I think the real problem that you're really (implicitly) getting at is that of the loud, outspoken fringe drowning out the silent majority. In public events involving completely *self-selected* populations, debates tend to be dominated by those with the strongest positions and enough time and motivation to devote to pushing their particular cause or agenda.

This is the key benefit of "public participation" events being organized so as to ensure that it is both accurately representative of the public (and not just the outspoken activists or special interests) and moderated appropriately to ensure that the "silent majority" voice is heard as well as the more extreme positions. In particular, inviting randomly-sampled participants rather than a self-selecting group, as in a citizens' assembly or deliberative poll, can ensure more that the debate is more representative and less biased, and that representatives of the normally-silent majority in the larger population will have a chance to speak on behalf of that majority in the "mini-public" assembly. Such a representative group certainly will and should include some skeptics, maybe even some vehemently outspoken on one position or another, but with appropriate moderation it becomes clear what the majority versus outlier positions are, and all the direct participants are forced to (and benefit from) learning from each others' positions and finding points of agreement and consensus despite coming from very different places. This is one of the well-known benefits of deliberation among a diverse and truly representative group: it can lead to not only better understanding of and consensus around the topic at hand, but also can help the *participants themselves* understand each other and the subtleties of issues and positions better.

Christian Folini @christian.folini · 5 months ago Maintainer

Time to wrap this up. Here is my draft summary.

Do you see other ways of motivating the general public to learn about and improve the security of the system?

Here are a few examples to start the discussion:

- *Run a hackathon to develop voting clients that facilitate testing of the servers*
- *Run a hackathon to write the main parts of the crypto code of the control components in a different language*
- *Run a contest for improving the current solution (e.g. for a more secure printing office)*
- *Create a contest for social engineering plots and reward the best ones according to some defined criteria (e.g most impact for least number of people fooled).*

Specialized technical events like a hackathon can introduce more people to the electronic voting systems or raise the interest in the technical community.

For the public at large a different approach is needed. Finding a good mode or channel in this regard is very challenging.

Citizen science projects, or extended workshops involving a random representative selection of participants comprising a citizens' assembly, could help address wider audiences directly and/or through the media.

It would also be interesting to make the voting system tangible for ordinary people or to find roles to participate in the electronic voting process for them like the vote counters on a municipality level. This is meant to bring democratic accountability and ultimately contribute to the public confidence into the system.

If you do not agree with my summary, then please shout. If you do agree, I welcome a confirmation either in writing or a simple upvote. In case there is no feedback or no negative one, I will sooner or later assume consensus and close this discussion.

[EDIT] Updated the citizen science paragraph on request of @Bryan.Ford below.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Looks good, but I would suggest clarifying the "Citizen science" paragraph as follows:

> Citizen science projects, or extended workshops involving a random representative selection of participants comprising a citizens' assembly, could help address wider audiences directly and/or through the media.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bryan. I think this is more precise and more elegant at the same time. Adopting it.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Looks good.

Christian Folini @christian.folini · 5 months ago Maintainer

No negative feedback, so I am assuming consensus with the slightly updated summary.

Thank you for your participation.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

8-risk-management-and-individual-risks.md 40.6 KB

Discussion Block 8 - Risk Management and Individual Risks

Risks management is a central element for the security of the internet voting system and its processes. As per art. 3 VEleS, by the means of a risk assessment, the canton must document in detailed and understandable terms that any security risks are within adequate limits. To do so, they are to address the security objectives described and by no mean minimising risks must be dependent on keeping security-relevant information on the system and its operation secret.

1. Organisation

The necessary knowledge of the system, conflicts of interest and comprehensive and effective risk coverage are the issues at the centre of the organisation of risk analysis in the complex context of internet voting in Switzerland. Consider the following actors:

Actor	Role	Reputation risks	Financial risks	Technical risks	Legal risks
Confederation	Legal requirements Authorisation	Yes	No	No	Yes
Cantons	Implementer	Yes	Yes	Yes	Yes
System provider	Cantons' contractor	Yes	Yes	Yes	Yes

As each of these actors has a different perspective on risks, each of them has to manage the risks it owns. Cantons and system provider also share most of the critical information assets but have separate infrastructures, thus weaknesses and strengths.

In order to solve this puzzle, several solutions emerged from the answers to the questionnaire, none with a majority:

1. The Confederation, in collaboration with the different stakeholders (which may also include representatives of the science, security experts, a committee of experts etc.), issues a core security concept that inventories threats, proposes mitigation controls in a comprehensive manner and defines the responsibilities for implementation.
2. As this task needs sovereign capabilities, the cantons are responsible for it. They can however get help from outside if they lack the competencies. (status quo)
3. A committee of experts fully independent from the confederation and cantons is created and in charge of doing the risk analysis.

It is quite difficult to imagine a committee of experts that is fully independent from the confederation and cantons. The question of who will define its mandate and effectively mandate it arises, who if not the Confederation or the Cantons. The second solution could be perceived as the same as the first one. However, they differ in the fact that, in the second one, the cantons are responsible for the whole risk management but in the first one, the different stakeholders agree upon responsibilities. In the current context, the first solution appears to be a better match.

Concept

A possible implementation of the solution 1 would define the responsibilities as follows:

- Confederation
  - Establishing a core security concept in collaboration with the different stakeholders
  - Establishing its own risk analysis and action plan
  - Reviewing submitted risk analyses and action plans
  - Monitoring its risks
- Cantons
  - Establishing their own risk analysis and action plan
  - Reviewing the system provider's risk analysis and action plan
  - Monitoring their own risks
- System provider
  - Establishing its own risk analysis and action plan
  - Monitoring its own risks

The process of risks management could be schematised according to the two following figures:

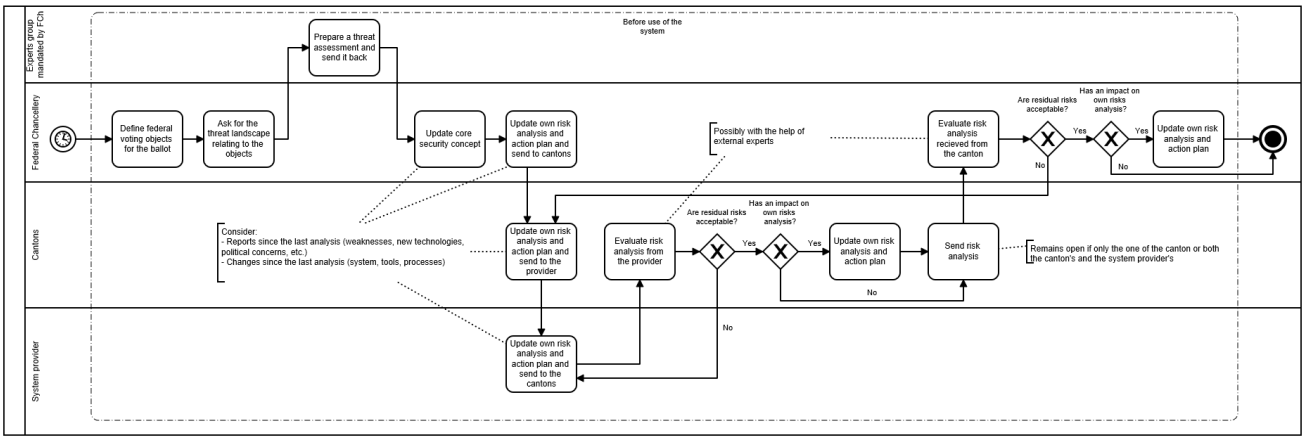


Figure 1 - Before using the system (before voting period)

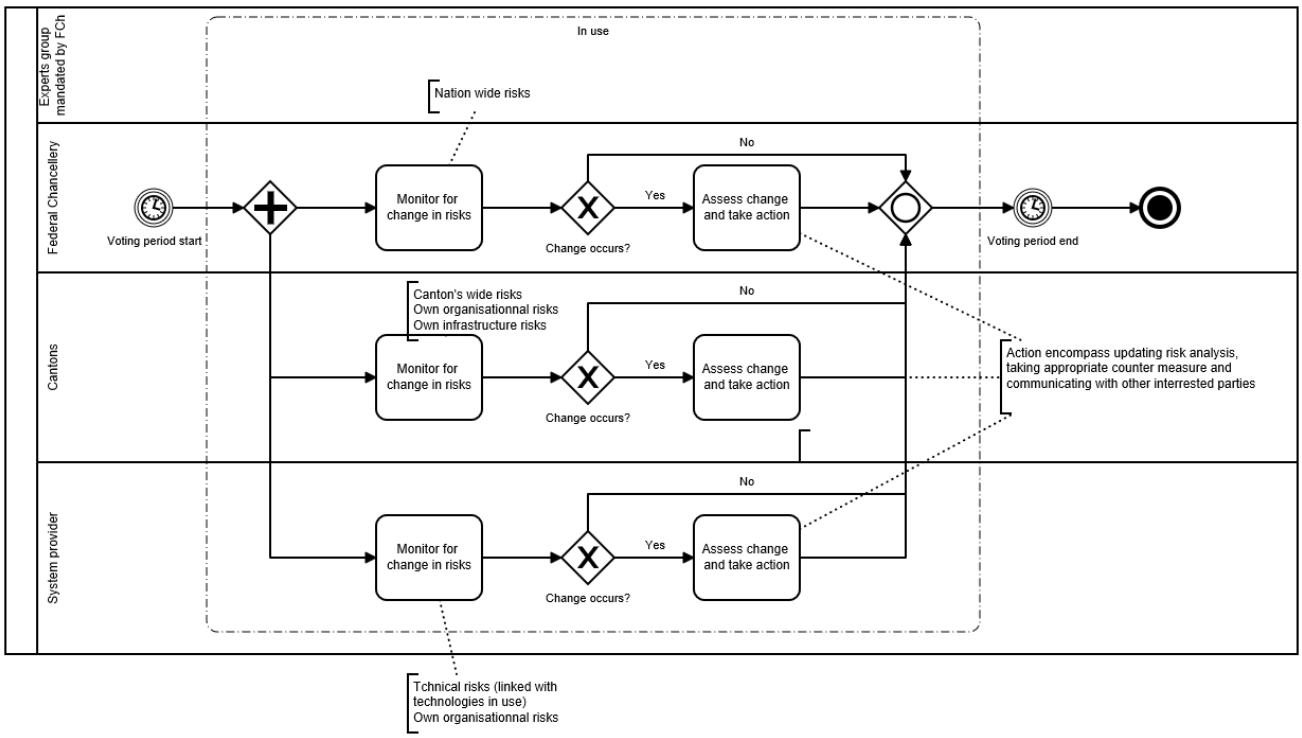


Figure 2 - While using the system (voting period)

2. Update

A regular update of the risk assessments is a requirement in the information security standards. However, recommended frequency of those updates looks like “regularly” or “whenever necessary”. The question here is what does “whenever necessary” mean in the context of internet voting in Switzerland. While some would define this point in time as whenever the system changes or whenever a new threat arises, the most relevant time in the context of internet voting in Switzerland is before each ballot as the object of the vote has an impact on the threat landscape. This review will only affect a part of the risk assessment, thus taking less time than doing a full review. However, enough time is to be planned for this activity, as some actions might have to be taken to mitigate updated risks. Voting usually takes place four times a year (every 3 months), this frequency can also be understood as regularly. As only a specific part of the risk assessment impacted by the ballot itself and events that have happened since the last ballot will be reviewed each time, it might also be appropriate to review the whole risk assessment once a year as per ISO 27001 prescription.

3. Methodology

The most important part in risk analysis is a comprehensive and systematic approach for analysing assets, threats, vulnerabilities and risks. As long as a methodology offers ways to achieve this approach, it can be used to asses an internet voting system’s risks.

Methodologies meeting these requirements could be:

- ISO 27005
- NIST standards
- OCTAVE Allegro
- BSI Grundschutzhandbuch

4. Publication

The publication of a risk assessment comes with benefits:

- Trust building
- Increased transparency
- Public involvement
- Allow wise decision (face the truth)

But also downsides:

- Revealing of weaknesses
- Controversy potential

The publication of the risk analysis requires resources, first of all for the formatting of it and the writing of the information allowing its better understanding. Secondly, to ensure that the questions and remarks that a publication generates are answered.

5. In practice

The VEleS already mentions a list of threats that are expected to be considered in the risk analysis. However, according to the answers we received, some threats seem to be underrated:

- Social-engineering
- Accidents / human mistakes
- Alleged attacks or malfunctions, framing attacks
- Planting trojan horse or backdoor

The state attacker threat agent is also mentioned to be missing.

Moreover, a particular attention should be paid to:

- Printing office
- Postal service (for delivering voting material)
- Zero-day vulnerabilities

In the answers given in the questionnaire, some specific risks have been suggested to be considered:

- Criminal organisation or foreign adversary infiltrates the trusted printing office (physically, socially, or electronically) to exfiltrate the codes on mailed voter cards and compromise cast-as-intended security.
- Criminal organisation or foreign adversary infiltrates the trusted postal service (physically, socially, or electronically) to misdirect some percentage of ballots from selected neighbourhoods to an alternate address where they are held or destroyed.
- Criminal organisation or foreign adversary offers a potentially large number of voters money or cryptocurrency in exchange for installing malware or spyware on their devices, which verify that the voters cast votes the way the adversary prefers (largescale electronic vote buying or coercion). An adversary could even carry out such an attack with almost complete anonymity with appropriate use of cryptocurrency and smart contract technologies.
- Compromised web server or App store serves a compromised version of the voting Web app and/or native apps to users.
- Compromised certificate authority, code-signing certificate, or developer signing keys used by an adversary to produce correctly signed but compromised versions of the voting Web app and/or native app to distribute to users.
- Network denial-of-service attacker prevents (targeted) users from casting votes electronically; forcing them to fall back to the mail or in-person process, in the expectation that many targeted users will give up and not vote at all due to the inconvenience.

6. Related Questions

The related questions are labelled The related questions are labelled Block-8.

6.1 Individual links to related questions

- Block 8 Discussion A - Organisation - One organisation, three alternatives
- Block 8 Discussion B - Update - Frequency and timing
- Block 8 Discussion C - Methodology - Standardisation
- Block 8 Discussion D - Publication - What, with which benefit and how
- Block 8 Discussion E - In practice - Particular risks
- Block 8 Discussion F - In practice - Mitigations depending on voters' support

7. Questionnaire

This block is based on the answers to the questions 2.2.1, 2.3.1, 2.3.2 and 2.3.3 as well as 6.1, 6.2, 6.5, 6.6, 6.7 and 6.8.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
2.2.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.3.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.3.2	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
2.3.3	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.2	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.5	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.6	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.7	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
6.8	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

8. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 8A - Organisation - One organisation, three alternatives (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Questions

How do you assess the assumption that the solution 1 (the confederation prepares a core security concept then all actors prepare their risk assessment under consideration of it) is a better match considering the context? If it is not an adequate one, which one would be a better choice and why?

How do you assess the concept of the solution 1 proposed in the organisation chapter?

Who should be responsible for assessing (possibly with the help of external experts) the coverage of the risk analyses from the cantons and system provider? Do you see a risk that there would be gaps between the coverage of the analyses?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to August 01, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

In my opinion, the concept of the solution 1 is a better match. Having a common security concept would favour homogeneity of the risk assessment process, would allow obtaining a consolidated picture of the e-voting system exposure to risks and possibly tangible elements for comparison between cantons if needed. The performance of the risk assessments themselves would be streamlined and shared feedback from the people performing the work could allow improving the method more easily if needed. Assuming that this security concept be elaborated through a collaboration between all concerned parties and based on consensus, I do not perceive any particular violation of the cantons' sovereignty. A common security concept would include common methodology, risk scale, threats catalogue, security measures catalogue (ISO27002, Common Criteria, NIST cybersecurity framework, BSI IT Grundschutz have been mentioned in the discussions so far, which is more than enough for the purpose). Also, the responsibilities in terms of information assets protection would be defined between the stakeholders, according to their respective role. The risk of gaps in the coverage of the analyses seems lower if such a consolidated approach is taken.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree that solution 1 seems the most promising, as it should help to consolidate and coordinate effort across the cantons, and enable some degree of sharing of the challenges and costs (financial and otherwise) of deep and challenging risk analysis problems.

A further question is whether an approach along the lines of solution 1 could be leveraged to make it more practical to coordinate and share the costs of proactive risk-management measures at system design and implementation time? As a specific example, could stronger design-time coordination at the federal level make it more feasible, logistically and financially, to implement some of the potentially-costly risk-mitigation measures explored in other discussion blocks, such as independent implementations of verification and control components? This may be formally out of scope of this block's "risk management" discussion, but I thought it might be worth revisiting in this context if only briefly.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Time is running out, so we better get going. Here is a draft summary of this question. I ask you to review it please.

How do you assess the assumption that the solution 1 (the confederation prepares a core security concept then all actors prepare their risk assessment under consideration of it) is a better match considering the context? If it is not an adequate one, which one would be a better choice and why?

How do you assess the concept of the solution 1 proposed in the organisation chapter?

Who should be responsible for assessing (possibly with the help of external experts) the coverage of the risk analyses from the cantons and system provider? Do you see a risk that there would be gaps between the coverage of the analyses?

The solution 1 is preferred by the experts.

The leading role of the Confederation in this setup allows for a homogeneous, consolidated picture that helps to compare the situation in different cantons. There is less risk of gaps in this approach than with the other proposed solutions.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

Christian Folini @christian.folini · 5 months ago Maintainer

I see three upvotes and no negative feedback. I am thus closing this discussion.

Thank you for your participation.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 8B - Update - Frequency and timing (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Questions

How do you assess the statement made in the update chapter (targeted update every 3 months + full update annually)? Would an additional review of the risks be useful?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to August 02, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

The proposed frequency is already challenging from an operational point of view. Risk assessments are lengthy by nature. Even if only targeted updates are performed every 3 months, we should not neglect the amount of work generated and hence the feasibility of working at such pace. Looking at things from the risk perspective, I believe that the threat landscape is not likely to evolve so rapidly that reviews need to be performed at a higher frequency.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree that risk assessments every 3 months seems aggressive, but perhaps feasible if at this period the risk assessments are strictly incremental and only identify and examine "on-demand" any significant new threats that might have appeared since the last 3-month cycle.

For more substantial periodic risk assessments in moderate detail, once per year seems appropriate.

Finally, I think that risk assessments with "maximum" depth and detail should be expected to coincide with the development and launch preparation of major new versions of the E-voting system, which might be expected to occur only once every few years.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Given the high frequency of votes in Switzerland, I think that a frequent risk assessment is in order, at least in the first two years. Then it could be scaled back based on a cost-benefit analysis.

Christian Folini @christian.folini · 5 months ago Maintainer

Here is a draft summary of this question. I ask you to review it please.

How do you assess the statement made in the update chapter (targeted update every 3 months + full update annually)? Would an additional review of the risks be useful?

The experts welcome a full annual update of the risk assessment and also a renewed targeted risk assessment every three months. However, they also agree that three months is somewhat steep and it might make sense to scale back to a less aggressive rhythm after a couple of years.

Major new versions of the internet voting system should also bring a full update of the risk assessment regardless of where they fall into into the update cycle.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Sharpening the sentence about the major software updates on request of @Bryan.Ford, upvoted by @Oscar.Nierstrasz; as well as a typo.

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

The last sentence currently seems so weak as to say almost nothing. How about: "Major new versions of the internet voting system should also have a full update of the risk assessment regardless of where they fall into into the update cycle"?

Also, nitpick: "every three month" -> "every three months"

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Bryan. This is stronger, but it was your statement anyways and I see Oscar supporting this proposal, so I have adopted it right away.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

looks good.

Christian Folini @christian.folini · 5 months ago Maintainer

I see two upvotes and additional favorable comments. I am thus closing this and thank you all for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 8C - Methodology - Standardisation (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Question

Is there a benefit in standardising the approach for all stakeholders or, on the contrary, would it be better to let each stakeholder chose the methodology he thinks is the most appropriate independently, provided that it matches the criteria mentioned in chapter 3?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to August 03, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

See response to 8A.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think a certain methodology "baseline" should be standardized and established at federal level, while leaving room for and encouraging stakeholders (especially the cantons and/or providers) to innovate and move beyond this baseline based on their own priorities and perception of risks and opportunities.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Here is a draft summary of this question. I ask you to review it please.

Is there a benefit in standardising the approach for all stakeholders or, on the contrary, would it be better to let each stakeholder chose the methodology he thinks is the most appropriate independently, provided that it matches the criteria mentioned in chapter 3?

The experts call for common security concepts that will result in a baseline risk analysis and homogeneity, that will allow to compare Cantons.

However, it would also be useful to leave room for Cantons to innovate and move beyond this baseline.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Two typos

Edited by Christian Folini 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Looks good.

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any negative feedback, but a positive response and an upvote.

I am therefore closing this discussion and I thank you for your participation.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 8D - Publication - What, with which benefit and how (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Questions

The publication of the core security concept would allow for public feedback on this topic. Is there a real risk that the publication could benefit a potential attacker, as he probably already knows about the weaknesses? Does it outweigh the benefit of the feedback? Would it be wise to publish further risk assessments (FCh's, cantons' or system providers')? On demand or systematically?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to August 04, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I do not think that public feedback is sought to discuss whether risks are acceptable or not, as "haters gonna hate". Therefore, I would not advocate for the release of detailed risk assessment results, which would prevent from disclosing potential entry points for malicious actions. A feedback by the public on the method applied would be valuable (e.g. are the threats and applicable security measures comprehensive?). For the sake of transparency, generic or consolidated figures could be provided to the public (e.g. risk rankings, number of vulnerabilities identified, fixed, newly discovered, etc., number and type of incidents endured, etc.). More details could be provided for vulnerabilities that have been fixed or for which a compensative control or workaround exists.

Edited by Stephane Adamiste 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

I agree with @Stephane.Adamiste that there may be justifiable grounds not to release fully-detailed risk assessment results to the public immediately. However, in the interest of transparency and public trust, I think it is important to release some information at an intermediate level of detail to the public, such as general findings and statistics as Stephane suggests. Further, transparency principles may suggest releasing fully- or near-fully-detailed risk assessments to the public after some time delay, after any important new findings have been adequately addressed for example.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Here is a draft summary of this question. I ask you to review it please.

The publication of the core security concept would allow for public feedback on this topic. Is there a real risk that the publication could benefit a potential attacker, as he probably already knows about the weaknesses? Does it outweigh the benefit of the feedback? Would it be wise to publish further risk assessments (FCh's, cantons' or system providers')? On demand or systematically?

According to the experts, the core security concept should be published in order to obtain valuable feedback.

The detailed risk assessment should not be published immediately for several reasons. However, it may be useful to publish consolidated figures or statistical information as well as a description of the governing body, the rules and formats for the publication of the detailed risk assessment.

A more detailed assessment could then be published after a reasonable delay, or when a new finding has been fixed, based on these rules.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Adopted a new wording after a discussion with several participants below.

[EDIT] Removed repeated "however".

[EDIT] Added "governing body" to the rules and formats on request of @Carsten.Schuermann.

Edited by Christian Folini 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think that saying "the detailed risk assessment should not be published for several reasons" is stronger than what I read either Stephane or me saying above. I would say something more like, "there can be justifiable grounds for not publishing the detailed risk assessment immediately"? To me that seems like a decision that should be made at each risk assessment: many specific risk assessments might (hopefully) be simply, "no significant findings", or "we found a couple things but are fully confident that they're not exploitable in the current operational configuration", etc., and immediate publication might well be justified in that case in the interest of public transparency when there's no perceived risk of doing so.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your take on this. I'm reading Stephane's statement differently.

I think it boils down to *publish by default* or *withhold by default*.

@Stephane.Adamiste: Would you mind sharing your opinion on this question here?

@Oscar.Nierstrasz has also been reading this so far. So maybe you want to chime in too.

Collapse replies

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

> I think it boils down to *publish by default* or *withhold by default*

I would say "withhold by default" detailed risk assessments as their content could provide insights to malicious actors. "Publish by default" stats, aggregated results, as discussed above. Release further details regarding detailed risk assessments once identified weaknesses are fixed, just as in a responsible disclosure process.

I think @Bryan.Ford 's statement would be more accurately reflected by just adding "immediately" in your sentence, i.e. "The detailed risk assessment should not be published immediately for several reasons".

I would add that rules and formats for publication should be communicated in advance to avoid suspicion concerning weaknesses being hidden.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I agree that "publish by default" should be applied except where this would introduce a significant risk. The formulation about by @Stephane.Adamiste makes sense to me.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Sounds reasonable, I agree as well.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you very much.

I have adopted the proposal about "immediately" and also added the suggestion by @Stephane.Adamiste about the rules and formats. When I did this, the whole response started to fall apart and I had to tie it together with several small edits also stretching into the subsequent paragraph.

Please check if this is still what you had in mind.

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

You wrote two times "However" in a row.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Stephane. Fixed.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I would suggest to add a comment, that clear rules should be established which body will take the decision to publish (when, what), simply to forgo the impression that the decision might be arbitrary.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Carsten.Schuermann . I have expanded to

> However, it may be useful to publish consolidated figures or statistical information as well as a description of the governing body, the rules and formats for the publication of the detailed risk assessment.

Christian Folini @christian.folini · 5 months ago Maintainer

I have not hear any additional proposals and no more negative feedback.

I am thus closing this discussion. Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Block-8 label 5 months ago

Christian Folini @christian.folini added Block-8 label and removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 8E - In practice - Particular risks (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Question

Do you agree that the proposed risks are relevant in the context of internet voting in Switzerland?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to August 05, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Yes, all of the proposed risks are extremely relevant (and there are no doubt others that we have failed to identify yet but will be identified in the future).

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I agree that proposed risks in the VEleS annex are relevant, but far from being comprehensive: Indeed, the list only considers human intentional threats (i.e. attacker's actions). It neglects accidental behaviour such as a bad manipulation or configuration errors by an administrator for instance. It does not consider breakdowns/malfunctions, vandalism on physical facilities, environmental hazards which could affect the availability of the electronic voting system. This is actually pretty indicative of the subjective nature of risk perception: People fantasize on undetected e-voting hacking attempts, whereas a breakdown at the ISP level preventing people to connect to the e-voting system is much more likely to occur (according to recent history) and is not even considered.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Yes, as has already been pointed out, most of the risks considered are those concerned with deliberate attacks, not accidental risks due to bugs, unforeseen subsystem interactions, usability gaffes etc. So they are all relevant, but certainly not comprehensive.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Here is a draft summary of this question. I ask you to review it please.

Do you agree that the proposed risks are relevant in the context of internet voting in Switzerland?

All the proposed risks are relevant and represent a good start, but the list is far from complete.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

[EDIT] Reworded in a slightly more uplifting way after input from @Oscar.Nierstrasz and the discussion below.

Edited by Christian Folini 5 months ago

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

"Far from complete" sounds too negative. I think it forms a very strong basis, but there are still some areas that could be elaborated.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for pointing this out.

How about this take?

> All the proposed risks are relevant and represent a good start, but the list is not yet complete.

Edited by Christian Folini 5 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

That suggests that at some point the list "will be complete", but that will never happen. How about: "All the proposed risks are relevant, but they should not be considered as being comprehensive, as new risks will always be uncovered."

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

@Oscar.Nierstrasz I do not think that this reflects the state of the current risk list. The list is currently not comprehensive as some risk categories are simply not considered, although being explicitly listed in the ISO27005 standard, which is supposed to be a reference. You pointed it out yourself..

Edited by Stephane Adamiste 5 months ago

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

OK, then let's stick with the last proposal by @christian.folini

Christian Folini @christian.folini · 5 months ago Maintainer

@Oscar.Nierstrasz: The way I read @Stephane.Adamiste 's comment, he agrees with the proposed summary (which he also upvoted), but not with my last proposal that took up your concern the summary was too negative. (-> *All the proposed risks are relevant and represent a good start, but the list is not yet complete.*)

@Bryan.Ford has also upvoted the summary, so I think Bryan and Stephane would both like to stick to that wording, even if it is fairly negative.

Can you live with that sentence, or do you really think it has to be a bit more uplifting?

@Stephane.Adamiste and @Bryan.Ford: Would you make a step in direction of @Oscar.Nierstrasz?

We could do this for example: *All the proposed risks are relevant and represent a good start, but the list is far from complete.*

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

I agree with the initial and updated versions of the summary. If we wanted to be more specific, we could write "All the proposed risks are relevant but the list misses risks that are not intentional human malicious acts"

Christian Folini @christian.folini · 5 months ago Maintainer

I'd rather not be specific since this implies the list would be complete if those acts were added (and I understood you all it's not).

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I like this last proposal.

Christian Folini @christian.folini mentioned in issue #60 (closed) 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Nothing to add from my side.

Christian Folini @christian.folini · 5 months ago Maintainer

OK. I have now updated the summary based on the discussion. As ever so often, there is sometimes more than one "last proposal" and I hope I really picked the right one. Please check.

Christian Folini @christian.folini · 5 months ago Maintainer

There has not been any additional feedback, so I think we're done here and I close this discussion.

Thank you for your contributions.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 8F - In practice - Mitigations depending on voters' support (Block 8 - Risk Management and Individual Risks)

Reference to originating discussion block

Block 8 - Risk Management and Individual Risks

Question

Part of the risk mitigation measures relies on the voter (individual verifiability, checking a certificate fingerprint, checking the hash of JavaScript files) and more of this kind of measures might come. Given the high frequency of votes, that proper training material is provided to the voter along with the voting material and that an easily accessible channel exists to report problems (i.e. Report button on the voting website), what would still be needed to ensure effectiveness of such measures?

Drop or upload designs to attach

Linked issues 0

Christian Folini @christian.folini changed due date to August 06, 2021 5 months ago

Christian Folini @christian.folini added Block-8 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Some of the measures we have discussed in the context of establishing and improving public trust - such as opportunities for anyone to participate in inspecting the system through hackathons, contests, etc., and for the non-technical general public to participate indirectly by supporting hackathon participants or through citizens' assemblies, for example - might also be effective in helping to train voters in using the E-voting system (verification and fingerprint-checking processes and such) and generally disseminate understanding of how the system works and should be used.

Educational materials in the form of instructional websites, tutorials, and videos can also certainly help - especially if tutorials are designed to allow "hands-on" experimentation (e.g., via Web-based simulations of the processes) that anyone can take part in, including those not directly eligible to vote. From an educational perspective, it is especially important for such materials to be accessible to children and students before they reach voting eligibility. This might eventually become a topic that should be covered at some point in standard Swiss educational classroom curricula, for example.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Usability tests are needed to ensure that the instructions are simple enough to be followed without error.

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

Existing measures towards end users already seem to provide adequate risk mitigation. Note that the issue reporting feature will prove inefficient in some circumstances (e.g. if a rogue e-voting site is presented to the user). Reminder popups during the voting process could be leveraged to encourage individuals to perform the risk mitigation measures they are responsible for. But again, this measure cannot be considered against the threat of a rogue e-voting site. One further step could be to ask voters to input results of their checks via an alternate channel. These extra tasks would however have a heavy impact on the user experience.

Collapse replies

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

At the risk of being repetitive, there is no substitute for live usability tests to verify that users can actually cope with the technology and the instructions.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Here is a draft summary of this question. I ask you to review it please.

Part of the risk mitigation measures relies on the voter (individual verifiability, checking a certificate fingerprint, checking the hash of JavaScript files) and more of this kind of measures might come. Given the high frequency of votes, that proper training material is provided to the voter along with the voting material and that an easily accessible channel exists to report problems (i.e. Report button on the voting website), what would still be needed to ensure effectiveness of such measures?

Risk mitigation measures relying on voters seem to be effective. Measures to establish and improve confidence into the voting system (namely block 7) are likely to bring a positive effect here as well. And usability tests will also contribute to this goal.

Education campaigns, namely for students and teenagers before they reach voting eligibility is being named as another useful initiative.

If you are not in agreement with my summary, then please leave a comment. If you do agree, it would be helpful if you could also leave a comment or upvote the summary. If there is no or no negative feedback, I will assume consensus and close this discussion.

Christian Folini @christian.folini · 5 months ago Maintainer

I see three upvotes and no negative feedback. I therefore close this discussion.

Thank you for participating.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

9-risk-limiting-audits-and-plausibility-checks.md 8.48 KB

Discussion Block 9 - Risk Limiting Audits and Plausibility Checks

1. Risk Limiting Audits

Risk Limiting Audits are a method to check the results of elections and votes. They provide statistical indications that the outcome of elections conducted with voter-verified and machine-scanned paper ballots are correct. See https://risklimitingaudits.org for a good introduction into the step by step process.

While designed for physical voting, the concept can also be used for electronic voting systems, where a paper trail exists (See Stark/Teague 2014). However, there is no paper trail with the online voting systems used in Switzerland. Yet several experts mentioned Risk Limiting Audits or the idea of a derived and adopted method in the questionnaire. It is therefore an idea worth exploring.

2. Plausibility Checks

Plausibility Checks are meant to detect any intentional or unintentional error with the entire voting process. For voting fraud, this is based on the idea that it is inherently difficult for an attacker to predict the result and tailor the fraud to blend in with the rest of the results.

Switzerland executes at least four national votes per year and depending on the residence of a voter also several votes on the canton and municipality level. During the years a very big pool of historical voting data has been accumulated. This could be a collection of information to draw upon when working with plausibility checks.

In Spring 2020, one of the largest voting frauds on record in Switzerland was discovered in the city of Frauenfeld in the canton of Thurgau. 100 paper ballots were misattributed to a different party. A simple plausibility check brought the con to light after the betrayed party formally complained. The ballots in question were quickly identified during the subsequent investigation.

Some sort of plausibility checks are established in most Swiss cantons, but they vary in the methods used depending on the administration.

In the responses to the questionnaire, several experts expressed doubts about the plausibility checks and especially cross-channel plausibility checks for various reasons.

According to those responses,

- there is a danger that low thresholds might lead to false positives that would undermine the trust in the whole system.
- statistical checks are not necessary, since they can not be used as evidence to identify fraud or problems anyways.
- individual and universal verifiability bring hard evidence and in the light of this evidence, the plausibility checks are not necessary.
- there is going to be a socio-cultural difference between voters using paper based and internet based voting, which will lead to different results on the different channels.

The idea of different outcomes across the different channels sounds convincing. Yet previous research shows that this is not necessarily the case (see Vassil 2016 for some insights based on Estonian elections). Maybe the socio-cultural differences and the differences between municipalities can be integrated into the statistical model: It is not readily understandable how gaps visible between different counting circles (there are more than 2000 municipalities in Switzerland) on paper would disappear completely when looking at the votes submitted via the internet.

Cross-channel plausibility checks (including comparison between and within municipalities) make it harder for an attacker to predict the results and to perform a fraud in a successful way. If the results have to pass the plausibility checks, then this forces the attacker to commit the fraud across different channels. This is more difficult and more expensive than the manipulation of a single voting channel. The introduction of voting channel diversity would therefore contribute to the attack resilience of the entire voting system.

3. Related Questions

The related questions are labelled The related questions are labelled Block-9.

3.1 Individual links to related questions

- Block 9 Discussion A - Risk Limiting Audits and other forms of audits based on sampling
- Block 9 Discussion B - Plausibility Checks

4. Questionnaire

This block is based on the answers to questions 4.12.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
4.12	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

6. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 9A - Risk Limiting Audits and other forms of audits based on sampling (Block 9 - Risk Limiting Audits and Plausibility Checks)

Reference to originating discussion block

Block 9 - Risk Limiting Audits and Plausibility Checks

Questions

Do you see a way to adopt the paper-trail based Risk Limiting Audits or a similar method to double-check the results of internet voting in Switzerland?

If yes, please elaborate.

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to September 01, 2021 5 months ago

Christian Folini @christian.folini added Block-9 label 5 months ago

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

Only if there's a paper trail that voters have checked accurately reflects their intention.

Bryan Ford @Bryan.Ford · 5 months ago Developer

It is a bit unclear to me what exactly it would mean to do a RLA on an e-voting election, and what concrete properties or benefits it would provide and what the results would mean. But it's interesting to explore that question.

By its nature, an RLA is really a process to verify with "enough" statistical certainty that a collected set of ballots was *counted* properly. It requires the ballots to exist in some retrievable way that can be accessed as an archive and sampled. But the part of the voting process that an RLA checks *starts* with those already-collected ballots, which the RLA presumes (hopes) have been gathered correctly, completely, that they reflect voters' true intents, etc. For example, with paper ballots, an RLA needs all the ballots to be either together in one location (in central-counting elections) or at least in a manageable number of locations (e.g., a registered pile at each district) so they can be sampled. An RLA won't do anything to increase (or decrease) the security of the process of a ballot getting from a voter into that registered pile, but once it's in that pile, the RLA helps ensure that the reported final count of all the piles is statistically reasonable.

As a mechanism to verify the *count* of a set of registered and archived ballots, therefore, the only way I can reasonably see a RLA working analogously in an E-voting process is basically to assume that all the (encrypted) digital ballots have at some point collected in some sort of (public or private) digital archives or bulletin board(s), then to sample and decrypt those ballots, and use the sampled count to double-check the outcome of the "master" count produced by the shuffle-decrypt-count process or the homomorphically-encrypted tally, whichever the E-voting system normally uses.

Viewed in this perspective, such an RLA mechanism might in principle serve to double-check that the shuffle-decrypt-count or homomorphic tally "worked correctly". This should of course be unnecessary if the verifiable shuffle or homomorphic tally protocols are correctly proven and properly implemented. I can see a potential argument, however, that doing a digital RLA to double-check the results of the cryptographic tally might be useful (a) just as an algorithmically-independent backstop in case the cryptographic tallying mechanism is fully broken but nobody realizes it, or (b) as a means to increase public confidence in the system even if with almost complete certainty the RLA double-check is technically unnecessary.

The cost-benefit of this approach to double-checking the tally would have to be considered carefully against alternatives such as those we've already discussed. The most immediately-relevant comparisons seem to be with the multiple-control-components and software-diversity discussions. If the E-voting system has multiple control components all running identical implementations of the same cryptographic tallying and proof-of-shuffle algorithms, then a bug in either the basic algorithms or their implementations might in principle affect all the control components together (i.e., a common-mode failure or failure of independence), and a digital

RLA of some kind might serve to detect that. Introducing diversity of implementation in the control component software would drastically reduce the risk that any single software implementation bug would compromise the whole tallying process, as we've discussed, but would still leave the risk that an unknown *algorithmic* flaw in the cryptographic tally or zero-knowledge proof could render all the control components - and hence the overall cryptographic tally - simultaneously vulnerable. A digital RLA, if done in a way that is algorithmically very different and independent from the way the main cryptographic tally algorithm works, could have the benefit of mitigating this residual risk.

However, this would be a fairly expensive mitigation for a risk that I think we're all generally agreed is one of the already least-risky parts of the system: namely a fundamental *algorithmic* flaw in the cryptographic tally mechanism, as opposed to an implementation flaw (or a client-side vulnerability, etc.). Incorporating the digital RLA would increase the E-voting system's overall design complexity significantly since it would introduce a major new and necessarily algorithmically-different design element in the system, not to mention the implementation costs. Not to say it wouldn't be useful - it might be worth considering! - but it would be costly and the residual risk area it would address seems to me to be far from the "low-hanging fruit" that realistic attackers will be looking for in the foreseeable future. It would also not address any of the known risks we've been discussing in terms of getting the vote correctly from the voter to the ballot repository or bulletin board: e.g., it would not help verify that any given ballot was cast-as-intended or correctly-encrypted or recorded (as opposed to being accidentally or maliciously dropped), etc.

Alternatives to consider: If it is deemed sufficiently desirable to have an *algorithmically-independent* check on the E-voting system's cryptographic tally, one alternative to a "digital RLA" might be simply for the E-voting system to implement more than one distinct proof-of-shuffle algorithm. There are quite a few different proof-of-shuffle schemes now available, a number of which are generally compatible in that they can operate on the same classes of ciphertexts (e.g., ElGamal encryptions in integer or elliptic-curve groups). So one might imagine the E-voting system's control components being upgraded so that each control component chooses one secret permutation, and one "before-shuffle" and "after-shuffle" pair of ballot lists, but then each control component produces *two* (or more) algorithmically-distinct proofs that these before-and-after ciphertext lists indeed represent a correct shuffle. This would ensure that if one shuffle proof turns out to be forgeable due to either an algorithmic or implementation flaw, the other proof(s) of the same shuffle would still protect the election's integrity. Unlike a digital RLA, there would be no statistical uncertainty: one would always expect the tally outcomes to line up exactly. If one flawed shuffle proof accidentally leaks the permutation, however, then voter privacy would be leaked: i.e., voter privacy would be protected only as much as the *weakest* shuffle proof.

A second variation on this idea would be for the "main" shuffle proof to be done using a perfectly-binding but computationally-hiding fashion, with the proofs verified only by the other control components due to long-term privacy concerns; and have other shuffle proof(s) with one or more distinct algorithm(s) designed to use be perfectly-hiding but computationally-binding (e.g., a shuffle proof over Pedersen commits). The latter shuffle proof(s) could then safely be released to the public without risking voter privacy either in the long-term, or in the short-term if the secondary shuffle algorithm's privacy proves to bet algorithmically flawed for some reason.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Cross-channel paper+digital audits: Upon further reflection, it strikes me that one potential attraction to RLAs in this context, despite their likely costs, is that they could conceivably work "cross-channel" to verify the counts across both the paper and digital voting paths. This would of course require all the paper ballots to have been recorded and collected in sampleable piles at one or more relevant locations, *and* all the electronic ballots to have been collected and recorded digitally in appropriate, sampleable tamper-evident logs and/or public bulletin boards or whatever. If designed correctly, in principle this mechanism should then be able to detect if a nontrivial miscount happened on either the paper or digital side (or both) that is sufficiently large to have a potential effect on the overall outcome, which would be a nice check to have. It still wouldn't be truly "end-to-end" in that it would not be able to check either cast-as-intended or recorded-as-cast properties on either the paper-based (postal) or digital voting channels, but it could at least cover the counted-as-recorded part of both the paper-based and digital voting channels together. So for that reason at least, this may be worth exploring further despite the cost it is likely to add. (And if it is decided to create a public bulletin board of some kind, a substantial part of the "digital RLA" implementation cost may be shared with that effort.)

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

As @Vanessa.Teague says. RLAs really only make sense, if there is evidence that is i) created and checked by the voter and accepted by the electorate as such, ii) it is secured with means trusted and accepted by the electorate between the vote casting and the audit. Paper ballots are to my knowledge the only form of evidence that is widely accepted (i.e. i), and in Switzerland it appears that authorities *are* trusted to secure the evidence during vote casting and before the audit (i.e. ii). There are very few trust assumptions needed to believe in paper ballots, and RLAs actually can reduce the trust assumptions further, by providing a way to identify and fix problems, both in paper only elections with manual counting and tabulation or paper ballots + digital interpretation and electronic tallying.

Encrypted ballots, digital signatures schemes, and zkpks are a technical attempt to create something that resembles evidence, but I would argue, that this form of digital evidence enjoys in no way the same levels of trust as paper does. There are many more trust assumptions (for example, verify the control codes) + lack of understanding, which in combination makes "digital evidence" vulnerable to allegations of wrong-doing.

Assume digital evidence is trusted, then recounting the evidence, possibly with alternative tools, is trivial. No audit (RLA) necessary. Assume digital evidence is not trusted, then not even an RLA will help to generate more trust.

In summary, and RLA is not designed to generate trust that the evidence (paper ballots, or digital evidence) is correct. It is designed to generate trust in that the result of the election is correct independent of the way how it was counted. (With an RLA you could guess the result and then audit it.) Therefore it is so important to strengthen the trust in the digital evidence, but an RLA is not the

right tool to achieve this.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you all for responding here and thank you Bryan to taking up a question where we did not expect much and squeezing some sense out of it.

I am a bit torn if that should make it into the summary. it's definitely interesting, but should it be in the summary? Reading the question anew, it explicitly states to think of alternative method to double-check the results. And I it Bryan's draws up possible alternatives in this regard. I am thus including it.

Do you see a way to adopt the paper-trail based Risk Limiting Audits or a similar method to double-check the results of internet voting in Switzerland?

If yes, please elaborate.

It seems unlikely that Risk Limiting Audits designed for paper ballots can be carried over directly to the current generation of internet voting systems.

Let's see if you agree with this. If I hear no feedback or no negative one, I am going to assume consensus and close this.

[EDIT] Adopted a stripped down summary when no consensus over additional answers could be found. See below. Remaining paragraph proposed by @Bryan.Ford .

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

@christian.folini . I think this summary needs a little bit more work. RLAs don't just verify the result of the correctness of the election, they will also fix it if it was wrong, by drawing bigger and bigger samples.

"It could make sense to strengthen the electronic mixing and tallying by developing different algorithms for the different control components." This sentence is very vague to an extent I don't understand it. What objective would this "strengthening" serve? To become more auditable?

"Risk Limiting Audits are a method to prove with enough statistical certainty that the evidence (ballots or receipts) printed on paper has been tallied correctly and will automatically correct a wrong result with a certain probability. Given the state of the art, it seems unlikely that the idea of RLAs can be carried over directly to the current generation of Internet Voting systems. More research is necessary to determine if it is possible at all."

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think one of the central questions here simply boils down to definitions: is a Risk-Limiting Audit *by definition* applicable only when the evidence (ballots) are in paper form? Or is a Risk-Limiting Audit a process to check (and perhaps correct) an election's outcome based on a body of ballot evidence (that's assumed to be correct), independent of what form that ballot evidence takes?

In the former case, then clearly RLAs are inapplicable *by definition* to either to Internet voting (or to multi-channel voting) where all (or some) of the underlying ballot evidence would have to be electronic rather than paper. But this inapplicability seems to be for a pretty shallow, and to me unsatisfying reason, namely just by saying we defined RLAs to be applicable to paper ballots only and thus it's applicable to paper ballots only. Trust us, we know, because we wrote the rule book that way.

@Carsten.Schuermann points out that "Paper ballots are to my knowledge the only form of evidence that is widely accepted", and I won't argue with that statement as far as it goes in terms of reflecting the current sentiment of most voting experts internationally. But as we've discussed extensively in this dialog, just because you've collected a big bundle of ballots in paper form and called it "evidence" doesn't automatically mean that nothing could have gone wrong in getting each voter's intent represented in (one and only one) ballot in that body of evidence.

Especially in a country like Switzerland where postal voting dominates, as Killer & Stiller analyze and we've discussed extensively, a lot of things could go wrong on the postal voting path a ballot takes from printer to voter, then back to the election authority - including not just ballot tampering but also chaos-sowing, denial-of-service, coercion, or (possibly-selective) voter disenfranchisement attacks of all kinds. And even a pure paper-based RLA will do nothing either to identify, or to correct, anything that might go wrong along that path from the voters to the bundle of ballots that we choose to call "evidence" for the RLA purpose. For example, an RLA will do nothing to detect that ballots were omitted from the evidence *and* the election outcome by being "delayed in the mail" for two weeks while traveling across town, and an RLA will similarly do nothing to detect that some ballots were improperly included in the evidence (and election outcome) after being purchased, filled in, and mailed by party operatives.

It's not that there's anything wrong with RLAs per se in this regard; it's just that everything that happens to ballots between the

printer, the voter, and the collected evidence is simply out of scope of what the RLA can actually audit. And taking an RLA to be an "end-to-end" check over the integrity of the *whole* election, as it sometimes seems to be portrayed, would be inaccurate - especially in a context like Switzerland where even the paper-based ballots arrived at the election counting office via a complex and highly electronically-dependent path.

So to me, a more meaningful and accurate definition for RLAs would be to say that an RLA is a method to take a body of ballot evidence - in whatever form - which the RLA simply assumes to be correct at the outset, and checks the election's final outcome against that accepted evidence.

By this definition, an RLA could in principle be implemented in a context where some of the accepted body of ballot evidence is on paper and some other part of the accepted body of ballot evidence is in electronic form (e.g., on a public bulletin board or blockchain for example). Either way, the RLA cannot and will not be able to check whether the ballots correctly reflected each voter's intent in that body of evidence, but it will ensure with high certainty that the final election outcome is consistent with that body of ballot evidence, whatever it is. Things can go wrong on a paper ballot's postal voting path from a printer to a postal voter to the election office's paper ballot evidence, and things can go wrong on an electronic ballot's bath from the control components to the voter's device and back to the E-voting authority's bulletin board or whatever for registration and tallying. But since those ballot paths from voter to body of evidence are outside the scope of what an RLA can audit anyway, why should an RLA - or why should the *definition* of an RLA - fundamentally "care" whether the ballot evidence it starts with happens to be in paper or electronic or stone tablet or any other form?

Perhaps one way around this issue is simply to try to state this definitional issue concisely in the summary. For example, working from the text @Carsten.Schuermann suggested:

> Risk Limiting Audits are a method to prove with enough statistical certainty that the ballot or receipt evidence an election is based on has been tallied correctly in the final election outcome. A Risk Limiting Audit will automatically correct a wrong result with a certain probability. While a Risk Limiting Audit can check the election's outcome against a body of ballot evidence, it cannot check the correctness of the evidence itself: that is, it cannot ensure that each voter's intent was correctly recorded and included in the body of evidence that the audit is performed against.
>
> Risk Limiting Audits are traditionally applied when the ballot evidence is in paper form. Some experts consider Risk Limiting Audits to be by definition applicable only to ballot evidence in paper form, and by this definition Risk Limiting Audits are inapplicable to internet voting. By a broader definition of Risk Limiting Audits as a process for checking an election outcome against a body of ballot evidence regardless of the form it takes, a Risk Limiting Audit could conceivably be applied to elections in which some of the ballot evidence is electronic. Such a Risk Limiting Audit would still be unable to check whether the (paper and/or electronic) body of evidence it is based on correctly reflects the intent of all voters. More research is necessary on the feasibility, usefulness, and operation of Risk Limiting Audits based on ballot evidence not exclusively on paper.

Edited by Bryan Ford 5 months ago

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

I don't think we can make up a new definition of Risk Limiting Audits - it is a thing that is already defined, and defined quite carefully, in the statistics literature. Here's Stark's definition from a USENIX paper from a while back:

> A risk-limiting audit has a guaranteed minimum chance of progressing to a full hand count if the apparent outcome is incorrect, thereby correcting the apparent outcome. The *risk* is the maximum chance that the audit fails to correct an apparent outcome that is incorrect, no matter what caused the outcome to be incorrect. Risk-limiting audits generally count votes by hand until there is strong evidence that the reported outcome is correct, or until all the votes have been counted by hand and the correct outcome is known. (https://www.usenix.org/legacy/events/evt/tech/full_papers/Stark.pdf)

The key property here is not whether the voters' intent is expressed on paper or not, but whether the computation can sample sufficiently to guarantee that it will not accept a wrong result except with probability bounded by the pre-determined risk limit. It has to be based on a representation of voter intent that you assume to be accurate, or whose deviation from accuracy you can assess arbitrarily precisely, or you cannot make the statistical argument that your audit process is risk-limiting.

So this is not applicable to any Internet voting system we know of, because we cannot assume that the electronic vote record accurately reflects the voter's intention. Even if we have a usable and effective verification mechanism, for example one in which 90% of voters reliably checked their return codes, we would not be able to perform a risk-limiting audit of a close election result to within a small risk limit, because no amount of auditing would give us valid statistical information about the intent of the 10% of voters who didn't verify.

I can imagine, in a society of perfectly truthful citizens, that it might be possible to perform a risk-limiting audit even on votes cast via a probabilistically verified platform, because the auditors could call up arbitrarily many voters and ask them how they had voted. But this is not feasible in practice, and anyway, even in Switzerland, people might lie.

So I think we should say, as @Carsten.Schuermann has said, that RLA's are not applicable to Internet voting because we don't have an accepted evidence base against which to audit the result.

(And @Bryan.Ford if you wanted to argue the same for postal voting, then I wouldn't disagree with that - we can't guarantee that that constitutes a carefully-secured ground truth either.)

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago

Developer

I don't think I disagree with your points @Vanessa.Teague about the technical properties RLAs have, but I think somehow we're talking past each other. The paragraph you quote from Stark's paper (a) doesn't actually read like a definition to me of what an RLA *is* but rather a paragraph summarizing certain (important) properties an RLA *has*; and in any case, (b) even if we do take this paragraph to be a definition of an RLA, I don't see anywhere in that paragraph (or anywhere else in Stark's paper) that it's stated or even suggested that the ballots audited must be *on paper*. Where did I miss that in Stark's (or any other "definitive") definition of what an RLA actually is?

You then say "this is not applicable to any Internet voting system we know of, because we cannot assume that the electronic vote record accurately reflects the voter's intention" - but this sounds like a quality/accuracy judgment about certain ballot-collection methods that reflects your own opinion (which I accept as perfectly valid and reasonable but it's your own opinion), and not something essential to the "definition" of an RLA either from Stark's paragraph you quote or another definition I've seen.

Yes, Internet voting systems often have trouble assuring that the collected and recorded (electronic) ballots accurately reflect the voters' intentions. But that has historically proven to be true in the case of paper ballots as well in practically all their designs and marking/casting mechanisms, even with in-person voting. Hand-marked paper ballots have all the well-known "voter intent" problems of figuring out whether this scribble was the voter trying to cross off a wrong choice and replace it with a right one, or was just an accidental marking... what did the voter mean if the ovals for two choices were filled but one more completely or darker than the other? etc etc etc. The hanging chads story. Then there's all the ballot-marking devices of various kinds and the many things that can and have gone wrong with them. Even if a ballot-marking device just produces a fully-paper "official" ballot, do you know for certain that "enough" voters actually carefully check all the markings or hole-punches or whatever on their paper ballot to be sure they matched what they entered on screen? Even if a ballot correctly reflects the choice they "intended" to mark, does that choice actually reflect their free will, especially if they took a ballot selfie in the ballot booth and posted it to FaceBook for their friends (and maybe their coercer or vote-buyer) to see after voting? These are just a tiny sample of the myriad ways even paper ballots cast in-person might, and historically have quite often, failed to "accurately reflect the voter's intention", even with in-person voting.

The point is that reasonable people may differ, and we could debate endlessly, on what types of ballot-marking, ballot-casting, ballot-collection, and ballot-storage processes do or don't provide adequate assurance that a collected set of ballots to be used as input to an RLA accurately reflects the voters' intensions (and especially whether it reflects the intents of *all* voters, given all the real voter disenfranchisement attacks we're seeing in practice). But whether some particular ballot-collection process is "adequate" according to your or my or anyone's opinion seems like a question completely orthogonal and unrelated to the definition of what an RLA in essence *is*. Even by Stark's quasi-definition that you quote, which says nothing I can discern about what threshold of acceptability is required for the collection of ballots to be audited.

I agree with your implication that the only way to make an RLA truly "end-to-end" would be to go back afterwards and ask a sample of voters how they voted - and somehow to (a) ensure that they can't lie but (b) also ensure that their vote records stay private, which sounds impossible in practice. Which just underlines the point I made a couple times above: in practice, a RLA is not an "end-to-end" check all the way from the voters' neurons to the election results. It's only a "late-pipeline" check that the election result is consistent with some particular body of ballot evidence, which is simply *assumed* and *hoped* to reflect voter intent accurately. Without perfect, privacy-preserving mind-reading technology, that assumption remains unverifiable with or without an RLA.

Christian Folini @christian.folini · 5 months ago

Maintainer

It seems settling this is more difficult than I thought.

Thank you for your various statements.

I will try and reconcile on a new summary. It may look as if you would be in disagreement, but I think we can find a consensus wording, when we stop calling the proposed electronic audit method RLA. I think this is also in line with the question that asked for RLA or a *similar method to double-check the results of internet voting*.

@Carsten.Schuermann pointed out, how weak my 2nd paragraph was. In fact I tried to summarize @Bryan.Ford 's proposal from above in a general way. But Bryan's latest comment brings a far simpler and more meaningful description and I think it is better to pick that up.

Here is a new wording of the summary. Based and inspired by your subsequent discussion:

> Risk Limiting Audits are a method to prove with enough statistical certainty that an election's ballot or receipt evidence has been tallied correctly in the final election outcome. A Risk Limiting Audit will automatically correct a wrong result with a certain probability.
>
> It seems unlikely that Risk Limiting Audits can be carried over directly to the current generation of internet voting systems.
>
> However, it could be useful to do research on the feasibility, usefulness and operation of a process for checking an election outcome against a body of ballot evidence regardless of the form it takes.

I have tried to keep it really simple and general. The idea is to If this research is actually launched based on this discussion block, then the detailed reasoning in this thread will certainly be taken into consideration too.

@Vanessa.Teague, @Carsten.Schuermann, @Bryan.Ford: Is this a compromise you can agree on?

Edited by Christian Folini 5 months ago

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

Well... I'm sorry to be a pedant here, but the key property of RLAs is not that they *prove* anything, but that they *don't* accept wrong results except with bounded probability. Technically, an algorithm that said, "do some auditing, but always be suspicious and recount the entire election manually regardless of the statistics" would be an RLA.

So I'd suggest removing the first sentence from the summary.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Seems reasonable to have the summary tiptoe around the definitional issue of whether an RLA can apply only to paper ballot evidence. However, to make the summary more neutral in that regard, may I suggest a slight clarification to the second paragraph as follows?

> It seems unlikely that Risk Limiting Audits designed for paper ballots can be carried over directly to the current generation of internet voting systems.

This way we don't have to decide or agree whether a Risk Limiting Audit can or can't apply to anything but paper ballots; the sentence remains true either way.

And maybe as a way to address the "prove" language @Vanessa.Teague objected to in the first sentence (which I think is useful but just needs to be clarified):

> Risk Limiting Audits are a method to verify with enough statistical certainty that an election's final outcome is consistent with the election's underlying ballot or receipt evidence. A Risk Limiting Audit will automatically correct a wrong result with a certain probability.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Vanessa.Teague and @Bryan.Ford for your subsequent comments. I am glad to read that we are making some progress on a consensus wording - even if we are not yet there.

@Vanessa.Teague: If we drop the first sentence, then we end up with "A Risk Limiting Audit will automatically correct a wrong result with a certain probability." And I think this is a bit thin as it does not even explain that this is about samling / statistics.

I see your concern with the word "prove" though, and I wonder if you would agree to Bryan's proposal that picks up your argument without throwing out the first sentence completely.

@Bryan.Ford: I think the addition of "designed for paper ballots" to the 2nd paragraph is a welcome clarification. If I do not hear any objections, I'm going to adopt this.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

First, again, let me say that I don't think we need to agree here. Sometimes it is an accurate summary of a discussion that there wasn't a consensus.

I'm sorry if I didn't make this clear, @Bryan.Ford, but the crucial property of an evidence trail for auditing has nothing to do with paper but relates to the exactness of the evidence as a way of demonstrating the election outcome. People could vote on shards of pottery or bronze disks, or compute an El Gamal ciphertext in their heads and check that it was accurately inscribed on a stone tablet. The medium doesn't matter - the crucial properties are accuracy and precision. As @Carsten.Schuermann said, it needs to be

> i) created and checked by the voter and accepted by the electorate as such, ii) it is secured with means trusted and accepted by the electorate between the vote casting and the audit.

The defining property of a Risk Limiting Audit is that it is *Risk Limiting* which is a pre-existing term in the Statistics literature, meaning that the probability of mistakenly accepting a wrong result is bounded. (Indeed, some US legislation specifies the risk limit in law, generally 5%.)

We have actually thought about this really hard in a variety of other projects, including the StarVote project and the vVote project (and Scantegrity too). In all these protocols (and also in the Swiss e-voting protocol) the opportunity to verify the cast-as-intended

property expires at vote casting time. Some people bother; others don't. This provides a probability of detection of error that is a function of the somewhat-random choices of those voters who bothered to verify. It might be sufficient to argue that a wrong result would have been detected with probability at least 95% (or whatever your threshold was), or it might not.

Here's the crucial problem: **if the voters didn't do enough cast-as-intended verification, the auditors can't go back later and redo it to increase their probability of detecting a wrong outcome**

So with all these systems, and every other electronic system I know of, you might in principle be able to do an audit (though I'm not sure what it would add to the cryptographic verification) but you can't do a *Risk Limiting* Audit because you can't go back retrospectively and do more checking. It's not possible.

[One can imagine, for future research, something that forced voters to verify or to record a transcript of their verification efforts, or record the randomness used in their ciphertext, so that if the election did turn out to be very close they could replicate the 'call up everyone and ask them how they voted' thought-experiment with electronic records, but I can't see how to do that in an appropriately privacy-preserving way.]

So does this answer your question? It's about arbitrary precision, not about the means of achieving it.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

So I would write something like, "unless the electronic vote-casting mechanism achieves certainty that the voter's intent has been accurately recorded, an audit of those electronic votes cannot be Risk-Limiting." And then you could add, if you like, something along the lines that some audits of some kind might be somewhat useful if there is further research. (Though to be honest I can't see why, but it's not incorrect to say 'maybe'.)

(And I know postal ballots go astray. I'm not defending vote-by-mail. I agree that conducting an `RLA' of mail-in ballots isn't really risk-limiting either.)

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Exactly, what @Vanessa.Teague said.

The reader of the summary may wonder if it is at all possible to design such an "electronic vote-casting mechanism that achieves certainty", and the verdict is still out. Hand-marked paper ballots can achieve this certainty, because the recording is immediate, but already for machine-marked paper ballots the situation is contested, because voters need to verify that their intent is correctly encoded. Some do check but others don't. I think it has been observed that the more races are on the ballot the fewer people check attentively. How to achieve certainty for electronic votes is unclear.

A related remark is that the "electronic vote-casting mechanism that achieves certainty" is actually not restricted to auditing but is also needed for trust-building in the counting process.

@christian.folini, if you make a proposal and integrate @Bryan.Ford and @Vanessa.Teague's comments into the summary, then we can look at it again fresh and as a whole for a final round of review. Thanks.

Bryan Ford @Bryan.Ford · 5 months ago Developer

@Vanessa.Teague I'm glad we're at least in agreement that whether the ballots are on paper or any other particular form is not what's important. And I'm very well aware of what the term "risk-limiting" means. But what you don't seem to understand - or can't accept for some reason - is that an RLA can *only* limit the risk of an incorrect election outcome *with respect to a particular body of ballot evidence*, which simply has to be *assumed* to reflect the voters' intent sufficiently well. Nothing in the RLA process can ensure that that's the case.

If you require that the ballot recording medium "achieves that the voter's intent has been accurately recorded", then I would contest that even hand-marked paper ballots do not reliably do this with "certainty", and thus conclude that RLAs fully satisfying your definition do not and probably cannot exist in practice. I think there is a lot of historical evidence to support this.

In any voting process, there are two security-critical "information paths": let's call them the *pre-sealing* and *post-sealing* paths. The pre-sealing paths are the N paths from each of the N voters' brains, through the ballots in whatever form they take, into the sealed body of evidence that the election authority collects at the election's conclusion and uses in both ballot counting and RLAs. The post-sealing paths are the paths from that sealed and accepted body of evidence, whatever form it takes, through the ballot-counting and/or RLA processes, to the announced election outcome. An RLA can audit - or statistically limit the risk of a failure on - everything that happens on the *post-sealing* path. But an RLA can do nothing to audit or limit the risk of failure on the pre-sealing paths from the voters' brains to the sealed ballot evidence.

And with in-person voting with paper ballots just as much for postal or e-voting, all kinds of things can go wrong and have gone wrong on those pre-sealing paths to interfere with voters accurately expressing their intent. Just some examples:

- A hand-marked paper ballot might be poorly-designed so as to confuse thousands of voters into filling in the oval for candidate C while intending to vote for candidate B. Due to the confusing ballot design, thousands of others overvote and hence have

their ballot not counted at all, because they first filled the oval for candidate C, then realized their mistake, but then filled in the oval for candidate B in addition instead of going back for a fresh ballot. Sound familiar?

If an RLA is done on this election, the election auditors might completely agree with the vote-counting machines on which ovals were filled - you can dial that statistical risk limit down to 0% if you like - and it will still neither detect nor correct a failure like this, because the failure happened on the *pre-sealing* path that the RLA fundamentally can't audit. The overall risk hasn't been reduced to 5% (or 0%); only the risk that the election's outcome is *inconsistent with the collected and sealed evidence*.

And with paper ballots just as much as for e-voting or postal ballots, the voter's ability to check or correct errors "cast-as-intended property expires at vote casting time": they "can't go back later and redo it later", even after hearing on the news after the election that thousands of others had apparently made this mistake. How many voters in this scenario would have loved to go back and double-check and perhaps correct whether their intent was correctly expressed, but couldn't because the election was already over and their ballots sealed? I don't see how this is fundamentally different in any way between paper (whether in-person or postal) and e-voting.

- In another scenario, thousands of voters intend to vote in-person and in fact show up to vote, but are told they're not allowed to vote because of a new tighter voter ID requirement, or because their voter registration was arbitrarily challenged or suspended and they didn't go through the bureaucratic process to resurrect it, etc. Sound familiar? These voters certainly *intended* to vote, and may in fact have been perfectly eligible to, but ballots expressing their intent were excluded from the body of ballot evidence before it was sealed: i.e., the failure occurred on the *pre-sealing* path that an RLA cannot audit or limit the risk on.

- During a pandemic, thousands of voters want to and intend to vote, but they're told that they are not eligible to vote by mail or by e-voting (if it exists), so out of fear for their own and their family's health, they stay home and don't vote at all. The intents of these thousands of voters have completely failed to be expressed in the body of evidence that is both counted and audited by the RLA, but the RLA will say that everything is "A-OK!" and have precisely zero percent chance of detecting and indicating that anything went wrong.

You get the idea; the list goes on. And considering the last point, I think there are increasingly-reasonable arguments that the very real (but undetectable and unmeasurable by an RLA) risk to election outcomes, due to disenfranchising voters who stay away from in-person elections out of fear for their health, may in fact be much greater than all the (again many, varied, and important) risks pertaining to e-voting systems that we've been discussing in this dialog.

In other words, even if it were given that paper ballots reflect voter intent with complete "certainty" when cast (a proposition I definitely do *not* accept), then the very *requirement to cast those ballots in person*, against the advice of personal and public health considerations during a pandemic, may make the extremely real and un-auditable voter disenfranchisement risk far worse a risk than anything an RLA could plausibly detect or correct.

Thus, I have to object categorically to any language that states or implies that RLAs limit risk in any *absolute* sense: again, they limit risk only *with respect to an accepted body of ballot evidence*, however well or poorly that evidence that evidence might reflect voter intent. And I also have to object strongly to summary language that simply takes it for granted that in-person voting on paper ballots "achieves certainty that the voter's intent has been accurately recorded". That remains an issue on which reasonable people may (and clearly do) differ.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for continuing this conversation. It obviously is not for a lack of trying that we fail to come to a positive resolution here.

This thread here is not a central part of the dialog, so it does not matter too much.

I was thinking about replacing the first paragraph with a reference to a definition of RLA in the literature or online. But actually, we do not need such a definition here and neither a reference.

So we can drop that completely.

The 2nd paragraph has been clarified by @Bryan.Ford and I think there was no objection to his wording:

> It seems unlikely that Risk Limiting Audits designed for paper ballots can be carried over directly to the current generation of internet voting systems.

I am thus going to adopt that.

Now the outlook on the useful research has been challenged, a proposal by @Vanessa.Teague to replace it was attacked by Bryan again, so this is to no avail.

If there was more time, we could try and sort it out. But I need to close this, so I think we simply have to drop it.

This leaves us with the 2nd paragraph. It has received clear boundaries now, which implies that a new generation of internet voting systems could lead to a different situation. That warrants new research. So the research is still there if only between the lines.

With this is am closing this question and thank you for your passionate participation.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 9B - Plausibility Checks (Block 9 - Risk Limiting Audits and Plausibility Checks)

Reference to originating discussion block

Block 9 - Risk Limiting Audits and Plausibility Checks

Questions

Do you think political scientists and statisticians can develop a statistical model that will detect hard to explain differences between multiple municipalities and between the electronic and the conventional voting channels? Can you briefly sketch such a model?

Would such a statistical method and its use contribute to the trust into internet voting?

Despite a large amount of historical voting data, there is relatively little data available for Swiss internet voting, namely for domestic voters. A derived statistical model might therefore be weak in the beginning. But do you think a growing share of internet voters and more data accumulated throughout the years would lead to improved models and solve the false positive problem in the long run?

In other words: Are plausibility checks something that could get better and better the more we use them?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to September 02, 2021 5 months ago

Christian Folini @christian.folini added Block-9 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

While plausibility checks do run risks of false positives due to statistical aberrations or cross-channel differences in voting behavior, I do believe plausibility checks could nevertheless be useful if appropriate models could be designed to limit these false-positive risks to a minimum even given limited existing data and hence uncertainty about potential cross-channel effects. Since this form of statistical modeling is far from my expertise area, however, I cannot comment on how difficult this would be or sketch a concrete model. This seems to be a question that should preferably be asked of political scientists and statisticians.

Nevertheless, I do at least find it believable that such models could be formulated, and that their uncertainties may be reduced and hence they might be made more precise with time.

Factoring out cross-channel effects: Finally, I can think of one way that cross-channel effects could conceivably be studied further and potentially mitigated proactively to some degree. A canton that allows both postal and Internet voting could pick a manageable random sample of eligible voters - perhaps similar in size to a polling sample but large enough to ensure adequate anonymity - and ask each of those in this sample population (a) if they would be comfortable in principle using either postal or Internet voting, and (b) whether they would be willing to take part in an experiment in which they are *assigned* randomly to use one or the other method instead of making their own choice. Then take the subset of this sample who opt-in by answering "yes" to both questions, and assign each at random to use either postal or Internet voting. The opt-in population participating in this experiment might be temporarily assigned to a separate "virtual district" for vote-collection and counting purposes rather than their usual local districts, so that their votes can be tallied and publicly reported separately from those of the rest of the population. The election results from this "virtual district", representing voters who were randomly assigned a voting channel, could then be (a) compared against the postal and Internet-based election results for the rest of the population to gather data on the prevalence (or lack thereof) of cross-channel behavioral effects, and (b) used in its own right to study voter preference with any biases from cross-channel behavioral differences hopefully factored out.

Even if something like this approach proved workable, of course, statisticians and political scientists with the appropriate expertise would need to be closely involved in the design and validation of the process. But perhaps this may be worth exploring further.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

I think we all agree that large, obvious modifications to many ballots could be detected by something like this, and often are. For example, Philip Stark has a nice report about why he thinks a certain DRE was misconfigured in a recent US election. (Four DREs were sitting beside each other in the same polling place, and all received a lot of votes - the first three went 60/40 D/R; the fourth went 40/60. It seems unlikely that Republican voters flocked to that particular DRE, and much more likely that there was an configuration error that swapped the options.) So there's nothing wrong with plausibility tests if we interpret them as checks for large, obvious, and presumably accidental errors.

But that shouldn't be confused with verifying a real election result against the possibility of deliberate, careful, surreptitious manipulation by an attacker who knows what statistical test will be applied. Remember that all statistical tests have some degree of confidence that's never perfect. You never get a certain result - at best you can say, "If our model of the ballots was correct, then we would have observed XYZ with probability p." Suppose the election result is close (and remember that all the contentious ones are, almost by definition, close). What if the tallies from your Internet-voting sample population are a little different from the control group? It's never going to be exactly the same - how different is 'different enough to be suspicious'? How do you conduct further examinations if you are suspicious? What if your sample population isn't representative anyway, for example because those who were ambivalent between postal and Internet voting lived closer to large towns (and therefore had slightly different political views or better Internet connections) than those who were only willing to vote by Internet? In those cases, you'd expect a systematic difference, perhaps quite a large one, even in your carefully-designed experiment (which is still going to be a lot better than taking the data direct without trying to randomise who uses which technology). In short, I am highly skeptical that a statistical test could be designed that could distinguish small manipulations from small genuine population differences, within the confidence necessary to verify a naturally-close result, against an attacker who knew what test would be applied.

Note that Risk-Limiting audits are a completely different thing, because they're grounded in a (presumed) secure and accurate representation of the votes. In practice, this usually means that when the result is close you need to do a lot of auditing, or even a full manual recount. Unless the plausibility check has an equivalent to this escalation option, I can't see how you could guarantee a reasonable level of confidence in a close result.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I am sure plausibility checks will get better and better the more research you put into them. However, statistical checks always use the past to predict the future with a certain probability. Therefore, there is never a guarantee that the model is accurate enough to reflect all the things it needs to reflect.

Statistical models have been used extensively to point out irregularities, but they do not provide a way to explain them. Walter Mebane has been doing a lot of work to identify electoral fraud in countries based on election data alone (for example on precincts etc) website which might be an interesting read here. But analyses based on statistical models, such as his, cannot identify electoral fraud, they can only point to electoral irregularities. To claim fraud, you'll have to investigate and find hard evidence. See also Council of Europe, REPORT ON THE IDENTIFICATION OF ELECTORAL IRREGULARITIES BY STATISTICAL METHODS]

I think the situation is similar for electoral trust (using the very simplistic view that P(electorate trusts) = 1-P(irregularities identified)). The statistical model can identify the lack of irregularities, but in order to build trust, one has to look at the evidence.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for sharing your opinions on this. This is my draft summary. Please check.

Do you think political scientists and statisticians can develop a statistical model that will detect hard to explain differences between multiple municipalities and between the electronic and the conventional voting channels? Can you briefly sketch such a model?

Would such a statistical method and its use contribute to the trust into internet voting?

Despite a large amount of historical voting data, there is relatively little data available for Swiss internet voting, namely for domestic voters. A derived statistical model might therefore be weak in the beginning. But do you think a growing share of internet voters and more data accumulated throughout the years would lead to improved models and solve the false positive problem in the long run?

In other words: Are plausibility checks something that could get better and better the more we use them?

Yes, it is possible to develop such a statistical model. The quality of the plausibility checks depends very much on the model and its inputs, which is expected to identify large scale tampering of ballots with a higher likelihood than surreptitious tampering on a smaller scale.

But it is unlikely that a careful and surreptitious manipulation on a smaller scale would be detected.

It is likely that said statistical model would improve over time.

Analyses based on statistical models can not identify electoral fraud themselves. They can only point to electoral irregularities and trigger an investigation to find hard evidence.

Please provide some feedback if you do not agree with the summary. Confirmation of the summary through comments and / or upvotes is very welcome too, of course. If I do not hear from you I am going to assume consensus and close this discussion.

[EDIT] Reworded first paragraph based on a proposal by @Carsten.Schuermann below.

Edited by Christian Folini 4 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

"It is possible to devise a statistical model to support plausibility checks to detect large scale modifications of ballots. But it is unlikely that a careful and surreptitious manipulation on a smaller scale would be detected."

It is not clear to me what the different between large and small is. Some techniques in plausibility checking use Benford's law (measuring the deviation of the digit distribution from what is expected).

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

I tried to express the following: It is possible to use a statistical method to come to the conclusion that something is fishy when the paper ballots lead to a 60% yes / 40% no result and the electronic channel to a 40% yes / 60% no. I consider this large scale.

But that it is unlikely that a modification of ballots leading to a 51% yes / 49% no result vs 49% yes / 51% no could be detected in a credible way.

If I understood you, Vanessa and Bryan wrong in this regard, then we need to find another wording that is correct.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I don't immediately see any problem with the current wording, but also don't object to a change if someone can suggest a clearer alternative. Clearly the threshold between a "large" and "small" manipulation is going to be a subjective judgment in which everyone's opinion may reasonably differ, and I don't see any reason the summary should try to go there. I think it's sufficient that there are "large" manipulations and "small" manipulations, for some reasonable definition of "large" and "small".

Christian Folini @christian.folini · 5 months ago Maintainer

Yes, in absence of said statistical model it is very hard to define small and large with any exactness. That's why I left it so general.

Edited by Christian Folini 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Ok, if the statistical model compares exclusively "multiple municipalities and between the electronic and the conventional voting channels" then large and small makes intuitive sense. However, if you use other factors into account, such as ethnic diversity, affluence, etc. then it become no longer clear.

Here is my suggestion for rephrasing. "The quality of the plausibility checks depends very much on the statistical model and its inputs, which is expected to identify large scale tampering of ballots with a higher likelihood then surreptitious tampering on a smaller scale."

Bryan Ford @Bryan.Ford · 5 months ago Developer

I'm fine with the rephrasing @Carsten.Schuermann suggests, except "then" should be "than" just before "surreptitious tampering".

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you for your proposal @Carsten.Schuermann. I think this is a very good wording. The text has been adjusted.

I had to edit it a bit, to make it fit the question, but the meaning is unchanged.

Also thanks for spotting the typo @Bryan.Ford.

There are no other open issues in this thread. I am thus closing this discussion and thank you for your participation.

Christian Folini @christian.folini closed 4 months ago

Christian Folini @christian.folini removed Last-Call label 4 months ago

10-forensic-readiness.md 16.8 KB

Discussion Block 10 - Forensic Readiness

While the VElecS and its annex contain requirements for the security of the system, they do not contain particular requirements regarding digital forensic and incident response further than maintaining logs.

Individual and universal verifiability are at the core of the system to detect manipulations but an efficient way to react on what they might uncover is now to be defined.

Forensic readiness is defined as the ability of an organisation to maximise its potential to use digital evidence whilst minimising the costs of an investigation (See Rowlingson 2004). In the specific case of internet voting, forensic readiness helps in detecting and investigating suspicions of manipulation thus increasing confidence in the outcome of the ballot.

The content of this block is based on the answers to the questions 7.1, 7.6 and 7.7 as well as interviews conducted with digital investigation professionals.

1. Forensic and data collection

A timely detection is crucial, thus an efficient monitoring is to be put in place. A monitoring could include what is happening within the network and servers (technical monitoring), feedback from the voters (e.g., lack of access or individual verifiability checks failed) and feedback from election auditors (perhaps based on the output of the "verifier" introduced in discussion block 2). However, when it comes to investigating a case, the most critical part is data. The data you have collected before using the system as a reference of a clean system, the data you collect while using the system to be able to trace the actions back and the data you collect after the use of the system as a state of play.

2. Data collection Swiss army knife

As mentioned above, data is the cornerstone of investigation. You have to gather a lot of data even though part of it might not be useful or even looked at in the end. It is difficult to predict which data is to be collected and which not. It highly depends on the system and on the techniques of the attacker. A way to check if you have collected the right data and the right amount of data is to simulate the investigation of an attack. Note that collecting everything possible can also affect the performance of the system, the storage you need and the number of people you need to analyse the collection. Thus, a balance is to be found. We have gathered here a collection of tools that could be useful in this context.

The monitoring of the voting system and its platform should at least encompass centralized and secured logs as per required in the VElecS' annex. However, logs are as good as what they contain. As the set of information that are to be logged highly depends on the system in use and its architecture, it is not possible to define it in a generic way and a specific analysis should be conducted for each system. Decisions need to be taken based on the risk analysis and reflected in it. It should be possible to track an event through the logs, thus they have to be consistent with each other (e.g. time stamp, event id, etc.).

Aside from logs, an integrity check should run regularly on all files of the system that are deemed critical, including but not limited to registry files and operating system files. However, these checks suppose that you keep a reference base and maintain it along with all updates you make on the system, which can be resource consuming.

A third tool in this Swiss army knife would be a forensic agent running on the servers. This agent would be able, for example, to take "images" of some parts of the server like its memory, its disks and its running processes. Some further tools, like malware research and detection tools, could help analysing those data and identify suspicious patterns in memory. The forensic agents are powerful but they open a new area for threats as they have high privileges on the computer they run on and even if they act in a read only mode, one could exploit a vulnerability in the agent to do whatever he wants on the server.

The use of out-of-band management equipment could also be leveraged in monitoring the system. As a security measure, it allows to cut the management of the elements of the system from the common network. You can also use it to get the configuration files and other data from the elements of the system that are linked to it without having those data going through the common network. As for the forensic agent, this kind of equipment is double-edged and can be abused if not properly secured.

Finally, network traffic capturing can also help investigating a case. In this case, the frontend and the backend have to be considered. However, having an all-time capturing might generate a huge amount of data, not to mention that someone should also be able to analyse those data. A compromise would be to have the whole system ready for network traffic capturing but only activate it when a suspicion of abuse has been detected by other monitoring systems.

All these tools also raise some concerns on one hand regarding the way they transfer their results to a centralized monitoring system and on the other hand regarding personal data protection. As for the first area of concern, networks and firewall have to be carefully set up and the "out of band" management has to be properly secured in order for an attacker not to be able to use the same channels to compromise the system. The second area

of concern is a bit trickier and depending on the data that are collected, a data protection officer should be involved to ensure a proper handling of data that could be sensitive like IP addresses. The data on which vote secrecy depends should be handled with a “secret” classification level, other data could be considered as “confidential”.

3. Related Questions:

The related questions are labelled Block-10.

3.1. Individual links to related questions

- Block 10 Discussion A - Forensic readiness and confidence
- Block 10 Discussion B - Criteria for forensic readiness
- Block 10 Discussion C - Minimal Swiss army knife
- Block 10 Discussion D - Effective monitoring and vote secrecy
- Block 10 Discussion E - Capabilities and limits of forensic tools

4. Questionnaire

This block is based on the answers to questions 7.1, 7.6 and 7.7.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
7.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
7.6	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
7.7	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

5. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 10A - Forensic readiness and confidence (Block 10 - Forensic Readiness)

Reference to originating discussion block

Block 10 - Forensic Readiness

Question

Do you think investigations based on a forensic ready system is likely to provide convincing explanations in the case where anomalies are detected or suspected (e.g. inconsistent control-components)?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to October 01, 2021 5 months ago

Christian Folini @christian.folini added Block-10 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think that forensic readiness, and investigations based on information collected from forensic mechanisms, can be valuable and reasonably safe to use only in what I called the "peripheral" infrastructure in 7A: namely network router, firewall, and monitoring infrastructure *outside* of the components that must be correct and uncompromised (to a threshold) in order for election integrity and voter privacy to be assured. Connections and packets flowing between the trusted components and the external world, for example, are part of the untrusted world and under the adversary's control from the perspective of the E-voting system proofs, and thus non-sensitive from a formal integrity and privacy perspective. Forensic readiness and tools deployed in this peripheral infrastructure may be extremely valuable to help detect and identify remote attacks or attempts at attacks, and to detect and address denial-of-service attacks or other performance or connectivity issues. These forensic mechanisms could take into account such information as IP addresses and traffic patterns, since this is information that all ISPs along the path between the voter and the election authority, as well as sufficiently-powerful (e.g., nation-state) adversaries, should be assumed to have access to as well anyway.

However, in general I feel that such forensic analysis mechanisms **must not** be used to expose or analyze the internal state within the trust-critical "core" infrastructure of the system, such as the control components. For example, forensic mechanisms that can capture the working memories of processes systems could expose the most critical cryptographic secrets in the control components, hence making the forensic mechanisms themselves a critical potential vulnerability area. Since forensic mechanisms and their operation cannot reasonably be expected to be made fully transparent and disclosed to the public, allowing forensic mechanisms visibility into sensitive trusted-component state would fundamentally undermine the premise or promise to the public that the control components are verifiably doing *only* what they're supposed to be doing and nothing more, and in particular not leaking critical secrets.

Even worse, if common forensic-analysis infrastructure is used to monitor two or more control components, then this would critically undermine the operational independence of the control components and create a serious common-mode failure/compromise vulnerability where an insider attack using (or successful hack of) the forensic facilities could extract critical secrets from all of the control components simultaneously, completely defeating both voter privacy and end-to-end verifiably (by leaking the secrets used in cast-as-intended verification). This risk of undermining independence and introducing critical common-mode leakage and compromise risks via the forensic mechanisms is, in my opinion, a risk far more serious than the potential analysis benefits it might provide.

If forensic mechanisms in the formally-untrusted peripheral infrastructure reveal indications of a potentially-successful attack attempt against trusted core infrastructure, then a reasonable response might be to disconnect and remove the suspect core component (e.g., control component) from operation, perhaps replacing it with a fresh ready hot backup device; immediately and perhaps permanently decommission the suspect core component; and only *then* enable any forensic mechanisms it may support such as memory snapshotting in order to inspect its internal state for signs of compromise or other anomalies. Such a decommission-then-snapshot process might also be used proactively, e.g., at randomly-chosen times on a randomly-chosen control component, as a way to detect evidence of any silent compromise that might have gone undetected thus far. However, this event

would have to be considered to compromise any cryptographic keys the component may contain, such as the control components' master keys, and hence may be infeasible to do safely during an election in the current E-voting system design. (A system that supported t-of-n operation where t < n, and/or a system that supported dynamic re-keying or re-sharing of threshold secrets, would offer more reasonable options in this respect, since a "tainted" control component's secret could be invalidated the moment it is replaced and the group's collective secrets re-shared.)

Another possibility that may be worth considering is whether state-inspection forensic mechanisms could safely be used on core infrastructure such as control components, provided the actual trust-critical code and secrets are encapsulated within a hardware-security container such as an Intel SGX enclave, and provided the forensic inspection mechanisms can snapshot and analyze only the "untrusted" operating system state *outside* of the trusted hardware enclave. Besides the implementation and operational deployment complexities involved, however, I am skeptical that even this approach should be considered truly "safe", since trusted hardware mechanisms like SGX have been repeatedly shown to remain vulnerable to numerous vulnerabilities such as side-channel leakage of secret data from the enclave to the "untrusted" surrounding environment in the system.

Thus, in general my default position is that forensic mechanisms are valuable and reasonable to deploy in the formally-untrusted peripheral infrastructure, but **must not** be used on the trust-critical core infrastructure.

Bryan Ford @Bryan.Ford mentioned in issue #55 (closed) 5 months ago

Bryan Ford @Bryan.Ford mentioned in issue #56 (closed) 5 months ago

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

I think that there might be a misunderstanding. Forensic readiness does not mean that all digital traces need to be recorded. Only those which are considered pertinent should be. Not all forensic tools need to be used either. Given an environment with identified threats, it really depends on which kind of digital traces are considered to be relevant to detect problems, frauds, malfunctioning etc. Built-in digital investigation tools make it easier not only to efficiently identify digital traces, but also to authenticate, classify, analyze, integrate, interpret and evaluate them, as well as their authenticity and integrity, in order to reconstruct the event of interest, understand its impact and take informed decisions. Forensic readiness adds forensic consistency in comparison to traditional logs. It makes events concealing much more difficult to achieve. My position is that well-chosen and adapted forensic readiness should be used all over the system, even on the trust-critical core infrastructure.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle created branch `52-discussion-10a-forensic-readiness-and-confidence-block-10-forensic-readiness` to address this issue 5 months ago

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

Problem detection and adequate responses are cornerstones for the robustness of the e-Voting system. In case of a security breach, an attack or some malfunctioning, it is important to precisely assess the potential impact in a worst case scenario. Traditional computer security measures need to be reinforced by strong forensic readiness in order to make the deep investigation possible and realistic, after a problem is detected. Digital forensic investigators should work in close coordination with CERT teams during or just after a problem has been identified. Doing everything to avoid problems in an Internet Voting system is not sufficient to reach trustworthiness, even if everything is verified by trusted parties and uses state-of-the-art technologies and protocols. Trustworthiness requires four conditions to be fulfilled simultaneously: (i) Everything (reasonable) has been done in a transparent way to prevent problems and is verified by trusted, independent entities (during system conception & development, and for system protection) (ii) It is recognized that problems are expected to happen anyway (iii) Any problem happening is highly likely to be detected (iv) Any detected problem is highly likely to be solved in an appropriate way

Forensic readiness means that the whole system has been conceived and developed from the very beginning keeping in mind that problems might/will occur (ii) and that efficient and trustworthy investigating tools must be built on top of the system itself (iii) and (iv). In order to efficiently investigate the Internet Voting system in case of an incident, forensic readiness makes a significant difference. Events leave traces. Forensic readiness means that the whole system is monitored permanently in order to record pertinent traces in tamper-resistant databases. Trace integrity and authentication are central. Forensic readiness should also guarantee the chain of custody of digital traces after they have been recorded. It requires full traceability (e.g. ledger-based traceability using blockchain technology ---see [Tamperproof timestamped provenance ledger using blockchain technology, DO Jaquet-Chiffelle, E Casey, J Bourquenoud - Forensic Science International: Digital Investigation, 2020]) and detailed trustworthy logs in order to investigate suspect or unusual events. Cryptographic hashes of logs information (made non sensitive) could for example be stored in public robust blockchains. This makes digital traces suitable both for handling incident response and for potential judicial follow-up in case the incident goes to court. Well-tailored forensic readiness has to be fully integrated on top of the system itself. Digital investigation tools must be available, ready to efficiently identify, authenticate, classify, analyze, integrate, interpret and evaluate digital traces, as well as their authenticity and integrity, in order to reconstruct the event of interest, understand its impact and take informed decisions. This allows a rapid access to valuable information in order to understand the issues and efficiently choose adequate countermeasures when reaction time is critical. Forensic readiness, combined with redundancy, also allows more in-depth investigation in order to reach a thorough understanding of the event, to assess its actual impact and to take required new preventive measures if necessary.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

There is never any guarantee that a post-attack forensic analysis can reconstruct the timeline of events. But forensics can discover of course things. In my experience, what makes election system forensics different from other system forensics is that the forensic analysis and its conclusions will become a part of the electoral process and therefore politicized. Those who favor the election result will try to discredit the forensic procedures, and those who contest it will bring more "evidence", that should be considered as part of the analysis, and if rejected, they will won't recognize the result of the election. There are plenty of examples out there, Kenya 2017, Bolivia 2019. Similar to statistical audits, in my opinion, forensics can only identify problems, but the proof that something went wrong or that would justify accusing a party of wrong-doing, has to come from elsewhere.

To answer the question. No, I do not think that investigations based on a forensic ready system alone are likely to provide convincing explanations in the case where anomalies are detected or suspected (e.g. inconsistent control-components).

Collapse replies

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

I would rather say (@Carsten.Schuermann) that there is no guarantee in general (“never any” might be too strong) that a post-attack forensic analysis can reconstruct the timeline of events. It might succeed in some cases. It often does partially. The same holds for traditional security measures too.

Any control mechanism (even traditional cast recount by hand) can be politicized in the scope of a hot political topic like election or voting systems. This does not prevent us to support these control mechanisms. Forensic readiness can have different levels of “readiness”. Usual log files approaches could be considered as the first level. Higher levels improve the quality (trustworthiness and traceability) of recorded traces. This brings more confidence in the results, especially in a sensitive environment.

Forensic readiness aims at harvesting what is a priori believed to contain relevant traces (pieces of evidence) in case of an attack, certainly not all traces. This diminishes the risk of having lost important traces that cannot be found afterwards (or would be very hard to recover). Moreover, forensic readiness records these potentially valuable traces according to forensic science high standards, and make them more trustworthy than a typical log file. Eventually, forensic readiness allows a faster analysis of recorded traces in case of an incident. It brings extra efficiency and trustworthiness. This is valuable when time is critical and the process highly sensitive.

What are the expected results? Forensic investigation can identify problems (I agree), but I respectfully disagree that it can only do this. Indeed, forensic processes are expected to do much more. They can eliminate (abduction reasoning) explanatory hypotheses that appear impossible (with respect to observed, recorded traces) and evaluate the relative confidence between alternative, possible explanations. Forensic science brings a scientific approach to evidence analysis, evaluation and interpretation. It support CERT teams in their (scientific) decision.

Claiming that “the proof that something went wrong or that would justify accusing a party of wrong-doing, has to come from elsewhere” is ambiguous. First we should agree on what “proof” means. From my perspective, a full proof (which is 100% sure) never exists. I prefer talking about strong evidence. The relevant strength of evidence to take a decision (threshold) varies from one phase to the other (it is lower during the investigation phase and becomes higher and higher as we approach the court phase). The strength of evidence threshold to take a judicial decision is the highest. To justify accusing a party or even to condemn it, it is often necessary to consider a body of corroborating evidences from different sources. Forensic analysis of the system if one important source among others. Forensic analysis alone might not be sufficient, I agree. However, other sources might become useless without a strong forensic analysis of the system to backing them up.

Forensic readiness has to be combined with redundancy to improve understanding of what happened and to evaluate the impact in case where anomalies are detected.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I think we may not so much be in disagreement, as talking about different categories of forensic processes and tools.

I have no objection to the use of appropriate tamper-evident event logging mechanisms (whether the tamper-evident log is a blockchain or something else), and in fact strongly support such tamper-evident logging - provided they are designed sufficiently carefully that these logs can never leak critical secrets. This has often been a problem in the past: e.g., numerous companies that thought they were managing password databases securely were later discovered to have been leaking plaintext passwords and the like into their logs. In the E-voting system, if the control components produce regular event reports that go into a tamper-evident log (and they should!), then exactly *what* events get reported and what information gets included in each of those event reports must be considered part of the trust-critical system design and must be scrutinized for accidental leaks or other issues just as carefully as the basic cryptographic design and implementations of the control components. And what gets reported and how must be hard-coded and determined in advance by the design and implementation of the control components themselves, and not dynamically determined by a remote forensics terminal that has the power to create new events or increase the richness of the information they report at will.

The kind of forensic tools I was referring to in my earlier comment, that I think should *never* be enabled on the trust-critical systems (e.g., control components), are the types that most definitely *can* leak secrets. For example, referring to the original question, a "Swiss-army knife" agent that "would be able, for example, to take “images” of some parts of the server like its memory, its disks and

its running processes." If some central forensics terminal, or its users, have the power to snapshot arbitrary parts of the memory of control component processes, then with almost complete certainty that forensics terminal could be misused by a remote attacker or corrupt insider to capture sensitive secrets such as voter choice codes or even the control component's master secret keys. Worse, if *one* forensics terminal has the power to perform such snapshots on *all four* control components, then whatever diversity and independence the control components might otherwise have is completely defeated by a successful attack on the forensics mechanism. Thus, any forensics mechanism supported - and *especially* one relying on any forensics infrastructure shared across multiple control components - must *never* have debugger-like powers of memory snapshotting or process manipulation, but only the power to (say) receive and record event records via logging mechanisms that have been carefully scrutinized for leaks as part of the control component designs, and that cannot be modified under the control of the forensics processes but only at design time.

Collapse replies

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

Thanks for the clarification.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

Thanks for the clarification.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for participating in this discussion and for trying to sort out your disagreements. I'm trying a summary and hope you can guide me to a reasonable and well balanced text.

10A got the biggest amount of responses in this block, some of the other questions were left unanswered. So I am integrating everything into a single summary response.

Forensic readiness anticipates that problems might or will occur. It adds forensic consistency to a system. Broadly speaking, forensic readiness is achieved via the careful integration of tamper-resistant log files and by the use of forensic tools that help to observe and analyse the operation of a system or its components.

Carefully planning the content of the log files and securing their creation, transfer and storage is a must. This applies to peripheral components that are not trusted in the system architecture as well as the core components of the system that are trust-critical.

It is important to note that no secret may be logged and that advances in cryptography could allow to decrypt encrypted information in the future.

The situation with the forensic tools and the out of band management is different: Their installation and use is welcome on the peripheral components. But it is critically important that none of the tools allowing to expose secrets are installed or running on the trust-critical components of the voting system: There always has to be a part of a voting system whose operations is unobservable so that vote secrecy and verifiability can be maintained.

It may be beneficial to limit the components that are allowed to run on a component via a hardware-attestation mechanism. This could be used as a proof that none of the tools that are dangerous to the security of the system are running on the individual components.

It may be worthwhile to decommission and snapshop a system that is no longer in operation or has strong suspicions of compromise in order to do a forensic analysis.

This is likely to be left open until the end of the discussion on the platform. Feedback is welcome.

[EDIT] Fix linebreak

Edited by Christian Folini 5 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We are getting to the end of this dialog very quickly now, so it's time to wrap it up.

I see an upvote and no negative feedback, so I am closing this discussion.

Thank you very much for your contributions throughout this discussion block.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 10B - Criteria for forensic readiness (Block 10 - Forensic Readiness)

Reference to originating discussion block

Block 10 - Forensic Readiness

Question

Do you know of a system that is fully forensic ready? What are the criteria for such a system? How should it be (publicly) scrutinised to ensure confidence in the results?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to October 02, 2021 5 months ago

Christian Folini @christian.folini added Block-10 label 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Sorry, I don't know.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We did not get any answers to this question. An overall summary for this block is being provided in the discussion of 10A.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 10C - Minimal Swiss army knife (Block 10 - Forensic Readiness)

Reference to originating discussion block

Block 10 - Forensic Readiness

Question

Which minimal subset of the above-mentioned tools would be necessary to properly investigate incidents? Do you see any critical tool missing?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to October 03, 2021 5 months ago

Christian Folini @christian.folini added Block-10 label 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

In my experience, integrity protection is not all that is necessary. Every file that goes into a forensic analysis needs to be attributable to a person or an authorized system that has created/modified it. Otherwise it is impossible to distinguish real files from fake files that may show up in the case of a serious electoral dispute. (Maybe this seems unlikely for Switzerland, but it has happened in other countries). Cryptographically secured logs are step in the right direction and an absolute must (finding a good solution for publishing the genesis log is mandatory). The cryptographic protocol should take care of the integrity / confidentiality part, so I believe that forensics will only become necessary in case availability of a system/services is attacked. As mentioned in 10A, the only forensic tools that should be used are those that produce reproducible and independently verifiable claims "that would hold up in court". It is difficult for me to assess which tools in the Swiss Army knife have this property.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We did not get a lot of answers to this question. An overall summary for this block is being provided in the discussion of 10A.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 10D - Effective monitoring and vote secrecy (Block 10 - Forensic Readiness)

Reference to originating discussion block

Block 10 - Forensic Readiness

Question

How could we set up the monitoring so that vote secrecy is still acceptably protected? Consider especially the data that are collected and the time during which they could be retained. Are there data that are useful for digital forensic and that would not hinder vote secrecy in the sense of how one voted (e.g. IP address)?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to October 04, 2021 5 months ago

Christian Folini @christian.folini added Block-10 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

See my answer to 10A. In particular, data (such as IP addresses) derived from the network or other formally-untrusted infrastructure that the security proofs assume to be under the adversary's control are in principle fine to use in forensics. Data in the trust-critical core infrastructure elements (e.g., control component state) are not.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

I thought the idea behind mixing is to protect vote secrecy, and to allow all other personal information, including IP numbers to be recorded and made available for forensics, even the encrypted vote could be recorded in a log file. The only issue I see is that the log files contain encrypted data that may be decrypted in 30 years' time. We have talked about this in an earlier question block on bulletin boards, and whatever we said there is relevant here.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We did not get a lot of answers to this question. An overall summary for this block is being provided in the discussion of 10A.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Discussion 10E - Capabilities and limits of forensic tools (Block 10 - Forensic Readiness)

Reference to originating discussion block

Block 10 - Forensic Readiness

Question

The forensic tools can have many capabilities (e.g. taking memory snapshots, monitoring running processes, looking into system files, imaging disks, etc.). Do you think that some of those capabilities should be limited to a subset on certain parts of the system (e.g. control components) and on which parts? On the other hand, do you see any capabilities that would be particularly useful on some parts of the system?

Drop or upload designs to attach

Linked issues 0

Christian Folini @christian.folini changed due date to October 05, 2021 5 months ago

Christian Folini @christian.folini added Block-10 label 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

See my answer to 10A.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Sure it should. If the forensic tool takes memory snapshots, which allows for the reconstruction of, for example, randomness used for encryption, when generating the private election key, or the permutation used during mixing and and decryption, then, yes, of course, this is not good. Terrible in fact. There always has to be a part of a voting system whose operations are unobservable. This is one of the trust assumptions, which raises the next question: How does one convince the critical public that no such forensic tools were running in the background?

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

Fully agreed. And responding to your last question,

> How does one convince the critical public that no such forensic tools were running in the background?

This is where I see a potentially valuable use for hardware-attestation mechanisms such as trusted boot or SGX enclaves. If everything that a control component is supposed to be running - including all processes' binary images, the OS image, the boot loader and perhaps even firmware images - is publicly known and hashed into a publicly-verifiable hardware attestation covering everything running from boot time, then that can serve as evidence the public (or at least any security experts that members of the public decide to trust) can use to get confidence that the control component is not running any unapproved forensics mechanisms with snapshot capabilities and the like. This would at the same time ensure that one of the public and approved binary images verifiably built from known public sources is actually being run, as discussed earlier in 6A.

Of course, I won't pretend that hardware-attestation mechanisms are perfect or unbreakable - they definitely aren't perfect and have already had their share of serious discovered flaws and weaknesses. But reducing the public trust required from "just trust our claim that we're running the correct binaries and no unapproved forensics mechanisms" to "the processor hardware certifies that we're running the correct binaries and no unapproved forensics mechanisms, and if that claim is not true then it means the processor [vendor] is compromised", represents a significant improvement in verifiable transparency.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We decided to take the block 10 together and provide but a single response for the complete block in the question 10A.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

11-big-picture.md 4.98 KB

Discussion Block 11 - Big Picture

1. "Big Picture"

The first internet voting trials were conducted over 15 years ago. The low-scale trial phase has allowed to learn and adapt as slowly more and more cantons joined in. Once the trials are resumed only a small number of cantons are likely to offer internet voting and that with only limited fraction of the electorate participating.

The expert dialog gives us an important foundation at redefining the trial phase commissioned by the Federal Council. Now that the discussions on the platform are almost over, we would like to ask you to relate your personal conclusions and issues you find important to the Swiss trials.

All experts are asked to share their thoughts.

2. Related Question

The related questions are labelled Block-11.

2.1 Individual links to related questions

- Block 11 Discussion A - The Big Picture

3. Questionnaire

This block is based on the answers to questions 1.1.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
1.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

4. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 11A - The Big Picture (Block 11 - The Big Picture)

Reference to originating discussion block

Block 11 - The Big Picture

Questions

How do you assess the current situation in Switzerland? Please distinguish between the level of requirements in the VEIeS (technical, scrutiny, transparency, ...) and the fulfillment of these requirements.

What would be the most important next steps and when should they be implemented?

Where would we ideally stand after the next 15 years?

All experts are asked to share their thoughts.

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to November 01, 2021 5 months ago

Christian Folini @christian.folini added Block-11 label 5 months ago

Stephane Adamiste @Stephane.Adamiste · 5 months ago Developer

The fruitful dialogue with all the experts involved in this discussion has shown that the level of requirements in the VEIeS is prone to improvements at all levels. This should in no way be interpreted as a failure of the e-voting initiative in terms of information security, as information security management is intrinsically a continuous improvement effort.

I would like to quote @Florian.Egloff 's sentence in his response to the questionnaire's initial question , as it perfectly illustrates my current thoughts: "Many of the questions that follow seem to be oriented in a techno-centric mindset, which assumes, that improving certification, using more secure cryptography, and getting more publicly verifiable notice boards will lead to more trust in the system. That may be part of the answer."

The next steps sould be, of course, to integrate the output of the topics discussed within this group into the design of the future e-voting ecosystem. In my opinion, emphasis should be put on applying rule #0 in information security management and communicating accordingly towards the public opinion: all decisions related to e-voting security shall be taken using a risk-based approach. The corollaries are that: (i) debates about building an inviolable system as a prerequisite to use electronic voting should be closed, as it is an impossible objective to reach; (ii) a risk assessment methodology that allows ranking e-voting risks in a comprehensive and rational way should be applied; (iii) objective criteria for residual risk acceptance should be defined; (iv) a degree of transparency towards the public regarding the various risks level should be adopted; among others, it would be very useful to explain that "presence of a vulnerability in a system" and "materialization of a threat scenario" are two different things.

Such an approach would: (i) allow defining priorities objectively for further improving the security of the e-voting system; (ii) demonstrate that the likelihood of hacking attacks leading to the successful manipulation of ballot results is overrated in people's mind given the robustness of the existing security measures, given the last 15 years of history, and given the fact that there are much more straight forward ways (in terms of feasibility and costs) to influence ballot results (e.g. using Facebook advertising campaign features). I believe that raising awareness about the true risk level would increase people's trust in electronic voting; (iii) break the vicious cycle that consists in always adding more advanced security features to fight the public opinion's scepticim, whereas it does not solve the problem: people will always be able to claim that the system is not "100% secure".

As an information security expert, I do not wish to depict an "ideal" situation in 15 years. It seems to me that the security aspects of electronic voting are only a subset of the requirements necessary to make such an initiative work as intended. The way democratic principles are enforced in a given country is a fundamental part, and I do not feel competent to discuss those aspects.

Edited by Stephane Adamiste 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago

Developer

Big Picture on Internet Voting in Switzerland – Florian Egloff

The first internet voting trials were conducted over 15 years ago. The low-scale trial phase has allowed to learn and adapt as slowly more and more cantons joined in. Once the trials are resumed only a small number of cantons are likely to offer internet voting and that with only limited fraction of the electorate participating. The expert dialog gives us an important foundation at redefining the trial phase commissioned by the Federal Council. Now that the discussions on the platform are almost over, we would like to ask you to relate your personal conclusions and issues you find important to the Swiss trials. All experts are asked to share their thoughts. How do you assess the current situation in Switzerland?

Currently, no internet voting system is available in Switzerland. For good reason, as the previous system was found to have substantial flaws. The federal council instructed the FedCH to work on a redesign of the trial phase with four objectives:

1. Further development of the systems
2. Effective controls and monitoring
3. Increasing transparency and trust
4. Stronger connection with the scientific community This dialogue is to be seen as to work towards that end. The FedCH states:

> "Participants in the dialogue should be able to discuss issues objectively, independently of political considerations, and without prejudging the outcome; the Federal Chancellery will therefore publish the results once the dialogue has been completed.» (https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting.html)

Based in part on my expertise in political science and cybersecurity, and in part on the participation in this active dialogue, I would suggest that triggering the dialogue was necessary. The dialogue showed the merits of including different sets of expertise, thereby sometimes opening up the pre-conceived notions underlying the questions, and discussing internet voting more broadly. Several requirements in the VEIeS were discussed and improvements identified.

I want to highlight here (for future dialogues and for transparency) that the majority of the dialogue, in my experience of it, stayed largely oriented in a techno-centric mindset, which assumes, that improving certification, using more secure cryptography, and getting more publicly verifiable notice boards will lead to more trust in the system. And I concur: this is an important part of the answer of how to build trust.

A socio-technical and socio-political analysis is needed

However, a socio-technical and socio-political view of voting would stress other factors, such as the social context, including political structures that have enabled trust in the current voting process, despite its (demonstratable) insecurities. Thus, under such considerations, other topics would be of relevance, including what political work is necessary to earn voters' trust? How does the internet voting technology need to be designed for and by the Swiss political process? How are voters' part of that design process? How is and/or should the system shaped by the localities, including the village level politics? I tried to raise these issues throughout the dialogue(see FN1). However, I do not think this dialogue probed these socio-technical and socio-political issues in sufficient depth.

To explain this, it is worthwhile to take a step back: one of the fundamental elements of democracy is the peaceful transfer of power through elections. In semi-direct democracies, referenda are the final arbiters of political conflict. Thus, it is key that the population has trust in the process and the legitimacy of its outcome. Any change that has the potential to disrupt this trust is per se to be looked at with some scepticism. Internet voting, in my assessment, has such a potential. I will make three examples:

Firstly, the risk of introducing doubts regarding the secrecy of the vote applies to internet voting in general, not just with regard to introducing a public notice board (as it was discussed in this dialogue) (see FN2). You can technically be more secure to guarantee the secrecy of the vote than the current system allows for, but socially introduce doubts regarding that new system. I would suggest any election authority introducing such a new channel would need a clear strategy of how you prevent that from happening, which would depend at the very least on having a clear mental model of which actions will contribute to voters being able to form trust in the new technology. This would also mitigate the larger risks of "turning away" from a technology completely that @Bryan.Ford raised in #19 (closed). However, the issue was not discussed in-depth in this dialogue.

A second example pertains to the inclusion of major stakeholders into the design, development, testing, introduction, and running of the new internet voting technology. Much thought and care should be taken into not just ensuring a transparent development and introduction of the technology, but also an inclusive one. If the technology is to earn the trust of the Swiss electorate, one way of forming trust and ensuring that the technology matches the society it is supposed to serve, is to include voters and other major political stakeholders (e.g. parties, media, election authorities) into the design of the technology itself. Furthermore, the introduction of such a new technology would need a strategy of how it would be introduced so as to enable citizens to understand and form trust in it. Such a strategy, however, was largely missing from the dialogue (e.g. the only item focusing on engagement with "the public" was in a sub-discussion of pentesting/bug-bounties, under the title "Other ways of public participation" (#37 (closed)).)

A final example is the absence of an in-depth discussion about crisis management and dispute resolution. An attacker attacking the legitimacy of elections only needs to attack the appearance of trustworthiness of the process or outcome, not the actual trustworthiness. If the elections can be made to appear to be "rigged" then the attacker has a chance to disrupt the fundamental value in democracy, namely, that the losers accept the outcome. Internet voting has the potential to introduce such a means of sowing doubt. This has again nothing to do with the objective "security" of the system, but rather, with the social familiarity and

trustworthiness of the system by the population: if the population does not have trust in the dispute resolution before the election takes place, it will be hard to convince a population of their "trustworthiness" in the actual event. For me, the key questions would be: Given an allegation of cheating/misconduct, is the remediation trusted and trustworthy by the population (voters, non-voters, non-eligible residents)? Given multiple threat actors deliberately injecting doubt and mistrust into and around the voting process, is the outcome still trusted and trustworthy by the population? However, there was no dedicated discussion about such strategies for crisis management and response, as a precondition for running an internet voting system in the first place. There are (to my knowledge) no clear, pre-defined and publicly communicated procedures for crisis communication for internet voting. Nor are there clear procedures for how voters will form trust in the proposed mechanisms for dispute resolution (see FN3).

Interaction effects with the existing voting system were out-of-scope for this dialogue

There is at least one issue, that was considered explicitly not addressed in this dialogue: a wholistic threat analysis was explicitly out of scope. I raised that capable threat actors that could act against the internet voting process could also act against the digital elements in the physical voting process. This means that treating those channels independently fails at addressing the threat wholistically. Consequently, risk mitigations should be taken in both, if one wanted to address the threat to election/referendum security (see FN4). @christian.folini provided space for discussion after multiple experts stressed the importance of the interaction effects of physical elements and internet voting to which I provided a lengthy answer (see FN5) concluding that: it is insufficient to rely on Killer & Stiller 2019 (see FN6) to assess how the existing vulnerabilities in the existing voting system can become a threat to election security once internet voting is introduced (and, though out of scope here [i.e. in my analysis], vice versa). Assuming that current mitigations to the current voting system would also mitigate threats to the internet voting channel's use of the current system is underappreciating the new risks that arise through the interaction of the two.

The issue is not purely an academic one, particularly if one considers the physical voting process as a fallback option (i.e. risk mitigation) for a potential failure of internet voting. Given a digitally competent adversary, the potential for disruption lies both in the internet and physical voting. Particularly, the risks to the process of voting should be seen as integrated (e.g. electoral rolls, printing, counting, vote tallying etc.). Consequently, significant efforts need to be spent to make physical voting more secure, before internet voting should be introduced. This may include for the federal administration to expand its regulations regarding the use of computers/digital equipment in the physical voting process. In my opinion, this should be within the scope of the discussion, when considering internet voting. In my personal opinion, I would encourage to first demonstrate mastery at managing the digital processes in the physical space before tackling internet voting.

The exclusion of these interactions between the existing voting system and the new internet voting channel, in my opinion, were unfortunate. Thus, going forward, in the spirit of a healthy public democratic discourse, I would encourage the Federal Council and the Cantons to try to be as transparent as possible about the existing risks and vulnerabilities in the current voting system. No voting system is ever 100% secure. All channels and processes have strengths and weaknesses. This would enable a more informed discourse about how and why internet voting is or is not secure enough to be introduced.

What would be the most important next steps and when should they be implemented?

The Federal Chancellery has raised the possibility of an academic advisory board. Based on the discussions in this dialogue, I see merit in such an endeavour. At all times, participants of the dialogue were civil, open, and neutral, and everyone brought their expertise to bear on the relevant issues. For an academic advisory board to be taken seriously, the participants have to be (as was the case in this dialogue) free to publish their contributions to the dialogue. Furthermore, they should explicitly not be deciding but should stay in an advisory capacity. I say this as the election system is a political institution, which means, that changes in such should be decided politically. It is not up to experts to judge which system is the best for Swiss democracy – we can advise, bring good arguments for and against, but not decide. Notably, such an advisory board should not only be a resource for the Federal Government, but also for the Cantons who are considering whether or not, and how, to go forward with regard to internet voting.

The dialogue has shown that, though internet voting is a highly technical subject, there are important technological choices to be made, and political units should get involved in deciding how those choices are made. The Federal Government should take care that it adopts rules that prevent a so-called race-to-the-bottom, e.g. prevent cantons from adopting less secure systems for financial reasons. This means, there is a responsibility for the Federal Government to mandate the development of minimum-security features and guarantees, should a service-provider want to offer internet voting technology in Switzerland. However, even when doing so, the cantons may still have a choice in how exactly to put together their mix of features in internet voting.

Where would we ideally stand after the next 15 years?

Ideally, there exists a shared cross-party consensus vision of how internet voting will become possible, with all major political stakeholders tied into the process. Politically, I believe the issue would profit from such politicization because as long as it stays in a more technocratic space, it risks being adversarially politicized. Thus, should there be trials, I would hope that the issue has moved away from an expert-driven / technical issue to a broadly understood one at the basis democratic level, with the major parties having agreed on how fast to move forward.

It would be my personal wish for Switzerland to chart a pathway, where one does not opt to introduce a technology before it is technically mature, whilst at the same time, not shy away from the political work of building a shared vision and roadmap of how one will be able to use digital tools for direct democratic participation.

Footnotes: FN1: Examples are: #10 (comment 384) #21 (comment 690) #24 (comment 694) #25 (comment 830) #25 (comment 866) #20 (comment 968) #37 (comment 948) #37 (comment 949)

FN2: See e.g. the following on why such a risk is important: https://www.cambridge.org/core/journals/british-journal-of-political-science/article/is-there-a-secret-ballot-ballot-secrecy-perceptions-and-their-implications-for-voting-behaviour/CD20CED9A450927354E8E754FC67D5E3; https://onlinelibrary.wiley.com/doi/full/10.1111/ajps.12019?casa_token=NN_KA8p67JIAAAAA%3AFpyCxhkwSZYtfsW7VxnJygaQjP5sMt1j4_zedd45lI2DhAJ_UM7eFAIMxZwf3igSOy1Tj322

FXoj5Wv-

FN3: It is insufficient to point to the “dispute freeness” of a protocol. Such a quality would still need to be explainable in-depth to a voter, before you actually run the system, and actual disputes will occur, as no perfect system exists.

FN4: #10 (comment 458)

FN5: #19 (comment 555)

FN6: https://files.ifi.uzh.ch/CSG/staff/killer/extern/publications/EVOTEID19-Killer-Stiller.pdf

Edited by Florian Egloff 5 months ago

David Basin @David.Basin · 5 months ago Developer

I focus below primarily on the question of how to design and implement systems to meet functional and security requirements, and to demonstrate this. Of course there are many other aspects to the questions below, not handled here.

How do you assess the current situation in Switzerland? Please distinguish between the level of requirements in the VEleS (technical, scrutiny, transparency, ...) and the fulfillment of these requirements.

The VEleS regulated different parts of the process and product. However, the scrutiny (assurance) requirements were insufficient, e.g., the relationship between the different evaluation artifacts (proofs, design, code) was insufficiently regulated. Moreover, there was limited transparency due to limited access to code, design, and evaluation artifacts. Finally, none of this was helped by the fact that the past product was fairly complex.

What would be the most important next steps and when should they be implemented?

Simplicity in the design should be strived for, but this is difficult to regulate in an ordinance.

A commitment should be made to using higher-assurance system development methodologies to develop key components. This is a nontrivial commitment that adds substantially to the development time as developing verified components is a substantial investment. Moreover, to succeed, it is best that the system is developed with verification in mind. I believe though that elections warrant this effort and care.

The goal of having verified voting components is achievable: theory is rapidly catching up with practice here. In the past decade we have seen verified operating systems, hypervisors, cryptolibraries, TLS implemetations, to name but a few. For example, the DeepSpec project (https://deepspec.org/main) is developing full-verified software and hardware including operating system kernels, cryptographic libraries and protocols, and even a networking server. These are "full stack" developments going from design all the way to code using the Verified Software Toolchain framework, which verifies C programs via a separation language embedded in Coq. The IronClad (https://www.microsoft.com/en-us/research/project/ironclad/) projects and IronFleet (https://www.microsoft.com/en-us/research/publication/ironfleet-proving-practical-distributed-systems-correct/) projects combine TLA-style refinement (correctness-by-construction development) with code verification. As a final example, Project Everest (https://www.microsoft.com/en-us/research/project/project-everest-verified-secure-implementations-https-ecosystem/) is building a completely verified HTTPS stack (https://www.microsoft.com/en-us/research/project/project-everest-verified-secure-implementations-https-ecosystem/).

Where would we ideally stand after the next 15 years?

Ideally we have the equivalent of a Project Everest in voting: where all the core components are verified and shown to meet their functional and security requirements. Of course this would still need to be complemented by other measures and activities to ensure that the right system is built, deployed, not modified, configured correctly, keys are setup correctly, etc. For example, one might additionally use a trusted execution environment to attestate the system so authorities can determine that the right system is running and protect against its attack by the OS.

Florian Egloff @Florian.Egloff mentioned in issue #58 (closed) 5 months ago

Reto Koenig @Reto.Koenig · 5 months ago Developer

We are very much in line with the other authors. But we do not only want to give thumbs up, but would like to share our thoughts on that topic:

We have been given the great opportunity to not only accompany the e-voting in Switzerland, but, to a certain extent, to form it as well for the past 10 years. We could see pure 'magic' and 'secret' e-voting, where political ideas were paired with economic views. Boosted with mystical 'cryptographic' ensnaring created the myth of 'everything is fine - don't you mind how it works' e-voting systems of the first generation in Switzerland. At this point, we were given the chance to help demystifying the cryptographic lulls and to clarify the true nature of the required 'belief' in the then established Swiss e-voting systems in order to gain trustworthiness.

Even though it seems that we were rather destructive at this time, it opened the path to a new Zeitgeist, where the focus shifted to transparency in running e-voting systems. Politics allowed academia to directly participate and form the requirements to the so called second generation e-voting systems. However, the system creators/vendors desired to blindfold academia during the hot phase of development (hidden implementation). So again, academia around the world was forced to demystify things in the

aftermath of development, way too late in the process. And yet again in the field of 'cryptography'.

Now, we are at the brink of the third generation e-voting systems, where academia is not seen as the opposing party anymore, but as an integral party *continuously* acting in a constructive way, in political, security and operational aspects.

As described by the other authors, academia is notoriously striving to the unreachable goal of a 100% secure system, where even the smallest step towards it is considered a great academic achievement. At the other side, the operators of such systems ask for a manageable system in terms of organizational and economical aspects. At political level, the main focus is to establish new means to maintain and facilitate the direct democratic process in Switzerland.

These seemingly contrary goals have one thing in common: The running system must establish enough trust as an alternative voting channel, so even the most surprising outcome of an election does not destabilize the democratic process in Switzerland. Thus, the system is not required to be 100% perfect, but perfect enough in order to prevail over the reasonable attacker scenario with a minimum of 'belief'.

If we allow all involved parties to jointly work together without seeing each other as 'the other side' but as 'a different aspect of the same side', hence, if we work towards the same goal all together, then, we at BFH see a practical e-voting solution being in place in 15 years from now.

More concrete, we see a system, that is directly derived from the political requirements, based on solid cryptographic- and organizational ground in order to provide a manageable and trustworthy system on operational level with a minimum of 'belief' required. The named parties *remain involved in a continuous process* of maintaining a running e-voting system that respectfully fusions the expertise, findings and requirements in politics, cryptography/security and operational power.

But then again, this only works if we all keep constructing, sitting and navigating in the same boat with the goal--not to be unsinkable, but--to remain manoeuvrable even in stormy seas.

Bryan Ford @Bryan.Ford · 5 months ago

Developer

As I expressed in 99A, one of the most important "big picture" points from my perspective is that given Switzerland's unique current situation, **the E-voting program must continue**. For better or worse, Switzerland is societally accustomed to and "locked into" the convenience of remote voting, and seems unlikely ever to be willing to return to the in-person voting methods widely accepted by international voting experts as being the most secure. But the currently-predominant system of remote voting by post hides many under-appreciated risks, including risks of large-scale hacking or selective denial-of-service attacks (e.g., disenfranchisement, biases, sowing chaos) against the vast amount of non-transparent electronic infrastructure embedded in modern printing and postal services. Postal voting and E-voting also share the same under-appreciated coercion risks. Most importantly, postal voting is a "dead-end" technology, which offers no readily-visible path to solve these security, privacy, and transparency problems. E-voting, while at the moment still far from as mature as we would like, offers the only plausible way forward towards a real solution that eventually could offer the convenience of remote voting but with security, privacy, and transparency comparable to in-person paper-ballot voting.

Although I expect this point may be controversial, I believe that the long-term systemic risks to Switzerland of being stuck with an increasingly-vulnerable postal-voting monoculture are much greater in total than the (many and varied) short-term risks from building and deploying a not-yet-perfect E-voting system among a limited user population while it continues to mature. Further, as I see it unlikely that the constituents of many cantons will likely be willing to provide robust, consistent, and continuous funding for E-voting development for years if the E-voting system is not actually deployed and usable by anyone (even a small portion of the population), I see the systemic risk to Switzerland of the E-voting program even being just "paused" as nearly as great as its outright cancellation, as there is too great a chance they would lead to the same end: namely being stuck for a decade or more with shaky dead-end postal voting technology. Therefore, the E-voting *must* continue producing usable systems, deployed and usable at least to limited populations, even if they are not yet completely mature in every respect we would like. We urgently need to fix the many security, privacy, and transparency issues we've been discussing, but in order to do so we must also successfully remain on a constructive long-term path.

I support most of the points made by others above, so I'll just summarize some of them from my perspective; namely:

- The E-voting program needs to be considered, developed, and evaluated, not in isolation but holistically together with the other voting methods in use, particularly including the predominant postal voting system.
- The voting program needs to develop systematic ways to involve the general public in the design and evaluation of future E-voting designs, both as a way to test and refine the usability in general, to ensure that security mechanisms such as cast-as-intended verification is *usably* secure, to develop materials and practices to explain E-voting to the public, and ultimately to earn the public's legitimate trust. Discussion 7D produced many excellent suggestions for this purpose.
- Coercion risks represent a major under-appreciated threat area affecting both postal and E-voting channels alike, and need to be addressed in future designs. This including large-scale vote-buying attacks, which can be done with either electronic or postal ballots, but also includes the numerous "coercion not to vote", i.e., disenfranchisement attacks in which an adversary might simply make the voting experience poor to discourage (perhaps selected) people or communities to give up on voting. Having to rely on a slow and unreliable foreign postal service may already be considered a serious disenfranchisement threat against Swiss citizens living abroad. Even if outright vote-buying is not currently perceived to be a critical or widespread problem in Switzerland, leaving the threat unaddressed and not even considered a goal leaves a major systemic risk and vulnerability area in the longer term that will be difficult to address quickly enough if and when it *does* become critical.

Next steps: I support the idea of maintaining an expert advisory board, similar to this one, but as a longer-term "standing" structure that operates either continuously or periodically on a regular basis. The membership should be rotating gradually to obtain a balance of institutional memory and fresh ideas and perspectives appearing regularly. The board should include academics from multiple disciplines, but need not and perhaps should not be entirely academic: it might be worthwhile for it to include some relevant industry and/or policy-community experts for example.

Just as importantly as the expert advisory board, however, I think the program also needs a "citizens' advisory board" representing the perspectives of the non-technical user population. Such a board could perhaps be organized as a citizens' assembly, for example, as I suggested in 7D. Just as importantly as its role of providing the experts and government better information about average citizens' perspectives, such a citizens' advisory board would help to train the *experts* and policy-makers alike how to make E-voting usable, how to explain what it is and how to use it in a truly accessible fashion, etc.

After 15 years: Needless to say, the E-voting program should continue addressing the security, privacy, and transparency challenges that we've discussed extensively here. Some of these challenges may not be realistically solvable in the next 5 years, but I see realistic hope that most of them can be addressed in the next 10, and definitely in the next 15 years. In particular, after 15 years I would like to see the E-voting system incorporate all the following protection measures:

- All trust-critical components (e.g., control components) have multiple, diverse implementations written in different languages and maintained by independent companies;
- All trust-critical code has formal *machine-verifiable* proofs of correctness both of the designs and the implementations;
- Multiple independent implementations of the client-side components similarly exist and support user-friendly cross-device verification of cast votes;
- A public bulletin board offers end-to-end universal *public* verifiability with long-term (at least post-quantum and ideally "everlasting") voter privacy;
- Strong coercion and vote-buying resistance across all voting channels: e.g., for ballots cast either electronically, by post, or in-person;
- Strong hardware-attested verifiability that the trust-critical components actually in operation are exactly those that have been made available to the public, in both source and binary forms;
- A robust body of educational and experimentation materials, developed and maintained cooperatively by both the government and the public, to support understanding of the system at all levels of technical (or non-technical) depth and build strong public trust;

Finally, looking beyond security/privacy/transparency - i.e., our concerns about all the things that might go wrong if we make mistakes - we also need to look at the positive *opportunities* that E-voting may facilitate in the long term, and work toward developing those opportunities.

In particular, Switzerland's tradition of four-times-a-year direct democracy also represents a highly successful experiment in making democracy more *participatory*. Many governments around the world at all levels have now adopted innovations like popular initiatives and referenda that were first widely-used here. But initiatives and referenda are just milestone and not the end of the path; Switzerland needs to continue innovating. Pushing and carrying out such innovation is not necessarily the role of an E-voting program, but the E-voting program should at least be conscious of and supportive of such innovations to whatever extent is feasible, such as at the cantonal and/or local government levels. The Swiss "federalist" tradition of allowing regional and local governments wide leeway to experiment and innovate has served Switzerland well for centuries, and the E-voting program should support this and especially not stand in its way.

There are many known ways democracies might potentially enable broader participation, beyond voting on a selection of issues four times a year. Ideas like liquid democracy, online citizens' assemblies, and deliberative polls are at the moment still experimental in various stages, but in 15 years may become mature and ready to be used at much larger scales. Switzerland's E-voting infrastructure needs to anticipate that future need and evolve (again gradually and cautiously) to support it securely. I believe this is possible, but it will require thinking ahead, and closely and regularly coordinating with cantons and local governments wishing to explore ways to make democracy more participatory and connected with the public.

Pursuing this participation objective will also help with the critical objective of earning the public's trust in the E-voting infrastructure. That is, offering voters real, perceptible value in being able to interact with their governments beyond four votes a year will give citizens a much stronger *motivation* to learn more about the E-voting system, its strengths and weaknesses, how they can use it most safely, and why they should trust it. Only a continuous, long-term, persistent investment in developing and improving the E-voting infrastructure - not only for better security, privacy, and transparency but also for better functionality and participation - will be able to achieve the program's long-term goals safely.

Edited by Bryan Ford 5 months ago

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I agree with most of what have been said in this thread -- a lucid assessment with, overall, considerable agreement.

Florian Egloff @Florian.Egloff mentioned in issue #8 5 months ago

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

It seems to me the role of regulations is to set a minimal set of standards and required processes, not to ensure that e-voting happens or doesn't happen. The ideal is to write standards that ensure that *if* e-voting happens then it achieves a certain set of desired goals (such as privacy, security, verifiability, transparency). At the moment, there may not be a system that meets even the current regulations - there's not much the Federal Chancellery can do about that. And I think many of us agree that the current standards could be improved and strengthened. Insisting on a higher standard (which is a good thing) inevitably makes that standard harder to meet.

I don't think we all have to agree. I don't agree, for example, with the statements above that the public tend to overrate the risks of the exploitation of security vulnerabilities, nor that e-voting will inevitably (or should necessarily) take the place of postal voting. But I think this dialogue has demonstrated the value of dialogue, just as our discoveries of cryptographic problems demonstrated the foolishness of secrecy.

I strongly recommend that the way forward is through an open process of both technical analysis and public discussion, with continuing leadership from the Federal Chancellery to define and clarify specific requirements. This forms a good structure for Swiss citizens to decide how (or whether) to continue with e-voting.

I hope this informed and thorough discussion is still ongoing in 15 years.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

Much has already been said. I would just like to emphasize two points:

1. There is no perfectly secure system. It is crucial that the public understand this point. You cannot eliminate counterfeiting. You cannot eliminate fraud. You cannot prevent people from getting fooled. You can only set up an efficient and effective system that (i) makes it very hard and expensive to cheat, and (ii) catches cheaters as soon as possible. With an eVoting system, you cannot eliminate by 100% the possibility of fraud, but you can make it too difficult and expensive, and you can catch fraud as early as possible. By establishing a strong track record, you can build trust.

2. The biggest risk is probably not actual security attacks, but rather trust attacks. We have already seen that many Swiss citizens do not want to install a Covid tracker app due to unfounded privacy fears. (Privacy risks are much greater with pretty much all of the other apps in existence.) It is not about technical competence or soundness but all about perception. One politician who claims that evoting will lead to massive voter fraud, can do much more damage than an actual attack.

Collapse replies

Bryan Ford @Bryan.Ford · 5 months ago Developer

Fully agreed on #1 (closed).

On #2 (closed), I don't think we need to our should try to decide whether actual security vulnerabilities, or the perception of vulnerabilities ("trust attacks"), are more dangerous or harmful. Both risk areas are extremely important and need to be addressed and mitigated. Taking a position that actual security vulnerabilities are more important could result in inadequate measures being taken to educate the public and create public confidence, as we've discussed at length and I fully agree on that. But similarly, taking a position that perception of vulnerabilities (trust attacks) are more important risks suggesting that policy-makers take the far-too-common approach in much of the world of simply trusting the security claims of some vendor pushing a proprietary technology, assuming "it's secure because it has a blockchain", etc. And over-emphasizing perception of security over actual security invites claims (true or not) that the system's claimed security is all show with no substance ("security theater"), which ironically can in turn undermine trust.

In other words, setting up "actual security" and "perception of security" as competing for importance against each other is counter-productive, because in reality they are both critical, orthogonal and complimentary dimensions. Neither must be neglected.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I fully agree. I just mean to underline that even the best technical solutions can be undermined by bad press.

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle created branch `57-discussion-11a-the-big-picture-block-11-the-big-picture` to address this issue 5 months ago

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

Trustworthiness requires four conditions to be fulfilled simultaneously:

1. Everything (reasonable) has been done to avoid problems and is verified by trusted, independent entities
2. It is recognized that problems are expected to happen anyway

3. Any problem happening is highly likely to be detected
4. Any detected problem is highly likely to be solved in an appropriate way

Prevention: system conception & development, system protection Doing everything to avoid problems in an Internet Voting system is not sufficient to reach trustworthiness, even if everything is verified by trusted parties (point 1). Being almost sure that there will be no problems is not sufficient either. It is important to face that problems are expected to happen anyway (point 2).

Robustness: problem detection, response The system should be robust against malfunctioning caused either by an intentional (internal or external) attack or by an unintentional incident.

Robustness means both (i) guaranteeing the availability of the system (e.g. the system is resistant to DDos attacks) and (ii) that any malfunctioning will be detected (point 3), and possibly corrected, with high probability (point 4). Detection of malfunctioning is crucial and highly challenging.

As for any sensitive physical system (nuclear plant, aeronautic, critical infrastructure, etc.), redundancy should be omnipresent in an Internet Voting system. Redundancy is crucial to handle point 3. Redundancy allows to discover some possible malfunctioning. Sometimes, redundancy is necessary for cross-verification. It makes many attacks much harder to achieve and often easier to detect. As a consequence, it diminishes the incentives for an actual attack and can therefore be considered as a prevention measure (point 1).

Once a problem is detected, it is important to be able to efficiently investigate the system and understand what happened in order to take care of the source of the problem (correct a possible bug in the system, understand and stop a discovered attack), assess the impact on the results, and draw further preventive measures if necessary. Redundancy has to be combined with forensic readiness to improve understanding of what happened and to evaluate the impact in case where anomalies are detected.

Forensic readiness aims at harvesting what is a priori believed to contain relevant traces (pieces of evidence) in case of an attack, not all traces. This diminishes the risk of having lost important traces that cannot be found afterwards (or would be very hard to recover). Moreover, forensic readiness records these potentially valuable traces according to forensic science high standards (chain of custody), and make them more trustworthy than a typical log file. Eventually, forensic readiness allows a faster analysis of recorded traces in case of an incident. It brings extra efficiency and trustworthiness. This is valuable when time is critical and the process highly sensitive. Forensic science brings a scientific approach to evidence analysis, evaluation and interpretation. It support CERT teams in their (scientific) decision.

Forensic readiness has to be fully integrated on top of the system itself. Digital investigation tools must be available, ready to efficiently identify, authenticate, classify, analyze, integrate, interpret and evaluate digital traces in order to help reconstructing the event of interest, understand its impact, as well as make informed decisions and decide which response is the most adequate.

I re-post my “Big picture” figure that summarizes what precedes:

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Thank you for an interesting dialog. Throughout this dialogue it has become pretty clear that modern cryptographic methods, program and protocol verification techniques, and improved security can all contribute to reduce trust assumptions and make Internet Voting more robust and trustworthy.

I think, however, it is a mistake to assume that just because one invests into these things, people will trust Internet-Voting more. (Conversely, I am quite sure that not investing into these things will make people trust it less.) A system that is otherwise secure can still be perceived as insecure by the public, a formal proof is good, but it might not hold up in the court of public opinion, and a verified implementation that runs in a production environment, I am not sure how close we are to that.

And then there will always be people crying wolf, disseminating accusations of voter coercion and the like that are accepted but beyond the reach of Cantons or the Bundeskanzlei's, or spreading allegations of corrupt election officials, computation errors, and cyberattacks. (Maybe this has not been a problem in Switzerland so far, but it might be in the future).

What would be the most important next steps and when should they be implemented?

1. Although we talked about RLAs in one discussion block, I was missing a more general discussion about how audits are being done. Are there processes in place to settle an electoral dispute with Internet Voting? What is required to conduct such an audit? Are the processes good enough to convince the critics, the public? I believe it is important to document these processes clearly in the regulation, to create a joint understanding between electorate and all stakeholders about what is acceptable evidence and how is it checked.

2. I have been always worried about backups and election system, especially, if the election secret key is constructed and leaked,

or reconstructed post-election. Any encrypted vote stored on a backup server can then be decrypted once someone with access is also in possession of the secret key, potentially threatening vote privacy. But I have considered this a solved problem, especially, because processes can be put in place to take care of this or never to reconstruct the election private key in the first place (for example, using partial decryptions). However, while participating in the discussion about forensics and forensics-ready systems, I found it highly problematic that a forensic software product, taking memory snapshots, can leak information about randomness or permutations, to anyone who has access to those memory snapshots. I think it would be good to address questions like this head-on and develop carefully worded requirements and public guidelines.

Where would we ideally stand after the next 15 years?

The Swiss electorate as a whole is proud of the Internet Voting system that the Cantons have procured and has confidence that it produces the correct result even though cyberattacks are likely to become increasingly prevalent. The Swiss voter population also understands the system so well that it can turn critics into believers. I believe, that the proliferation of knowledge about the Internet Voting system, how it works, and how it can be audited, is the best way to create public trust and acceptance.

Collapse replies

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 5 months ago Developer

I agree with you that trust and trustworthiness are two very different concepts. Having one of them, we cannot take the other one for granted. Trusted systems are not necessary trustworthy, and trustworthy systems are not necessary trusted! However, trusted systems which are not trustworthy are more likely to lose their reputation as a consequence of their internal flaws. Lost reputation would be hard to recover, especially in the context of eVoting. Therefore, making the system trustworthy is a must. But it has to be supported by trusted people, as well as backed up by transparent and honest communications to reinforce trust.

Srdjan Capkun @Srdjan.Capkun · 5 months ago Developer

I would like to pick up on a point that Bryan raised. We looked into voting in Switzerland some time ago and it is also riddled with risks. https://arxiv.org/abs/1906.07532 This is usually underestimated and people do assume that since 'it has worked so well so far, it is probably fine'. But increasingly even paper voting relies on electronic means.

So I fully support continuing electronic voting but one needs to make it using simple tools that are well understood, and the more complex the technology, the less acceptance it will have.

We have seen this also with SwissCovid app. The protocol is fairly simple but still generated a lot of discussions and took tremendous effort to make it happen. The fact that we could explain the protocol in simple ways definitely helped. https://ncase.me/contact-tracing/

Christian Folini @christian.folini mentioned in issue #60 (closed) 5 months ago

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

I want to briefly put forth two issues which have already been touched upon in the discussions; these issues are also reflected in some form by @David.Basin and @Stephane.Adamiste but I believe they are worth further underscoring.

The need for real-world data (next steps) -- or "don't let perfection be the enemy of the good"

Some of the discussion required significant speculation (and imagination). For example, we had to consider what attacks state actors would be able and likely to enact, how would the public understand different components of the system like the public board, or what is the likelihood and what is the approach we should take to deal with quantum adversaries.

This type of exercise is important and needs to be pursued. We should acknowledge however that a quasi-complete adversarial model, or an exhaustive list of challenges in terms of how the public perceives and interacts with the system will remain a theoretical exercise that may turn out to be divorced from practical realities. Attacks against a deployed system will identify and exploit novel attack vectors, (parts of) the public will misunderstand and raise trust issues which cannot be foreseen without real-world data from practical developments.

In my opinion it is important to embrace an agile approach where we deploy imperfect systems, learn from the ensuing experience, adapt the regulatory framework and re-design the system, and re-deploy. I would therefore caution against hard requirements which would lead to strong dependencies on standardization processes or, worse, require untenable advances of the state-of-the-art. This would hinder progress by introducing bottlenecks which are difficult to control, and would delay the collection and understanding of other key domain specific data.

Instead, I would advocate for a cautious approach which carefully balances the risks that are assumed with the potential impact on public trust (e.g. through best-effort security guarantees, exhaustive risk analysis and mitigation, full transparency).

**Formal methods **

Formal methods play a critical role in the development of robust software in a number of areas like hardware design, aviation and space programs, but lag far behind in the development of internet voting in particular (and in the area of cryptography in general). Formal method techniques require significant investment (as they are expensive both in terms of required expertise and amount of work involved) and in some sense, their use is also unrewarding (large amounts of tedious work that relies on well-established techniques to “only” get higher assurance).

I think an interesting challenge in this space is to design incentives which would help change this state of affairs; and to me there is no obvious path forward, and at least there is no clear parallel between the different application areas.

In those areas mentioned above, software quality has commercial/reputation but ultimately monetary impact for the owner of the product. In most cases the owner is also the developer of the product. Therefore, investment in formal methods can be internally quantified and justified and financed.

In contrast, for internet voting this type of relation/structure is missing: the developer and the owner of the system are rarely the same entity, breaks of the system are difficult to quantify, and thus there is no clear driver for the necessary investment.

Some obvious directions do not seem to work. Regulation cannot simply mandate the use of formal methods (this would be difficult to scope and difficult to specify); commercial entities are driven by … commercial incentives and it’s unclear how to structure these to be conducive to a more prominent place for formal methods in system development. Finally, academic involvement does not scale well to industrial settings and it is unclear how to effectively wove such an involvement within the actual development of a production-grade system -- past attempts where such a development of internet voting systems was left entirely to academia failed.

Privileged position for Switzerland

Switzerland is in an unique position to pursue and push the boundaries of internet voting. It benefits from a public that is educated, passionate and engaged with voting exercises. It also helps that such exercises occur quite often. Authorities benefit from a high level of public trust (fully justified by the general level of governmental transparency and accountability as well as the principled, scientific-based treatment of various issues). This state of affairs guarantees (to some extent) that efforts into developing effective internet voting solutions can continue. I particularly appreciate the current round of discussions as a first step in this direction.

It is impossible to foresee where internet voting will get to in Switzerland in 15 years -- a lot depends on unpredictable factors like political will, level of investment, developments in the cybersecurity arms race etc. At the very least, I hope that the two issues outlined above have been closed, that is, i) we will know how to balance the deployment of real-world internet voting while managing public expectations ii) we find a way to push the use of formal methods techniques used in high assurance software development to the field of internet voting.

Edited by Bogdan Warinschi 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

We have seen responses from 12 different experts for this question and given the response from BFH represents multiple people, we have extremely good coverage.

Writing a summary is still tricky, though, since this was not so much a discussion as a list of personal point of views. Which is very inspirational, but a bit challenging when trying to find the common ground.

When writing the draft summary below, I tried to concentrate on those ideas that were picked up by other experts or ideas that were mentioned by multiple participants in a similar form. If you read the draft summary carefully, then you will notice, that I also list a public board as a useful addition. However, a public board was only mentioned once in this thread here (mentioned in a positive way, that is). I am still including it, since it was featured prominently in several responses to the questionnaire and no discussion block brought a more passionate discussion than the one about the public bulletin board. Feel free to comment on that below and if there is a strong call to remove it from this summary, I'm of course open to do that.

Considering the number and size of the comments and my role to make a selection, it is quite obvious, that my gut feeling is strongly playing into the summary. Certainly more than with any of the other questions in this dialog.

So if the summary for this big picture is meant to carry any weight, you need to read it carefully and vet my ideas / help me expand it into the right direction.

Here we go:

How do you assess the current situation in Switzerland? Please distinguish between the level of requirements in the VEleS (technical, scrutiny, transparency, …) and the fulfillment of these requirements.

What would be the most important next steps and when should they be implemented?

Where would we ideally stand after the next 15 years?

All experts are asked to share their thoughts.

This is the summary of the Big Picture shared by the experts in the discussion block number 11.

For the experts, this dialog marks a milestone. The government authorities have invited scientists to discuss several pending questions around internet voting for the first time in an official process. This dialog has been very fruitful and should serve as a start of an ongoing exchange.

The experts identified several areas where the currently available internet voting systems are lacking or where the current Swiss regulation is prone to be updated in terms of security, privacy, verifiability, transparency and scrutiny.

The experts have addressed many problems that should be tackled immediately. Bigger technical challenges can only be implemented in a mid- and longterm perspective. They include the creation of formally verified voting components, more diversity and redundancy of the trust-critical system components and to use simplicity as a basic design principle.

Another challenge in the mid- and longterm perspective is a Public Bulletin Board. Whether and how a public board should be introduced requires further analysis.

Yet the dialog has also shown that a lot has been achieved during the past 15 years of internet voting trials in Switzerland. There is no need to abandon all future plans with internet voting. Security is a continuous improvement process and this dialog is a single step on this path.

For the experts, the dialog focused on technology to a very wide extent. The question of trust and how a society could start to trust a new technology remains to be discussed in the future. Raising doubts about the trustworthiness of an internet voting system or the results of a vote is one of the biggest residual risks with internet voting: One politician who claims that a vote has been manipulated can do as much damage as an actual attack.

It is very important that the design, the development and operation of internet voting does become more transparent. Yet it also has to become more inclusive at the same time. It has to include political stakeholders as well as broader groups of the voting population to find more acceptance.

According to the experts' big picture, the dialog focused on internet voting and left the existing physical voting channels aside. However, the existing voting channels are also prone to attacks; not the least via the auxiliary electronic system components. A holistic view on the possible attack vectors is thus necessary. This should lead to a holistic effort to improve voting security with a risk-based approach.

After all, there is no perfectly secure voting system and the voters should have a transparent view on the known weak spots with the voting process. You can not rule out fraud completely nor can you rule out technical errors. But you can make it very hard and expensive to cheat. And you can prepare yourself to be able to detect an attack as soon as possible. By establishing a strong track record, you can build trust.

Some experts expressed the necessity for strong dispute-resolution procedure that require the Internet Voting system to produce broadly accepted and immutable evidence that can be independently checked to settle claims of malfunction and cyber attacks.

If the experts look 15 years ahead, then internet voting is established and the population has confidence that the voting system produces the correct results. For some of the experts, said process will take a bit longer. For other experts, it is much less clear that Internet voting will earn the trust necessary for being established.

The next 15 years are therefore key since many problems are still unresolved. But this dialog has shown viable or at least potential solutions to many if not most of them. Continuing the dialog with subject matter experts as well as representatives of the population is thus a useful and promising method to move forward.

This summary is likely to be left open until the very end. Your feedback is equally welcome and needed. I hope we can settle on a balanced version and we can then close the dialogue together with this discussion here.

[EDIT] Rewordings throughout the document to rule out ambiguities and for better readability. No change of meaning with the exception of "scrutiny", that was added in the 3rd paragraph. It is true, that the term was only used by @David.Basin but many, many experts described something that is best summarised with this keyword. So I think it belongs into that list.

[EDIT] Separated the Public Board into a separate paragraph. It is not on equal footing in this conversation here as the other items in said list.

[EDIT] Several language fixes on request of @Bryan.Ford. "much more damage" replaced with "as much damage".

[EDIT] Added a sentence that explains there are several experts that don't agree that the establishment of internet voting is inevitable on request of @Vanessa.Teague.

[EDIT] Added a small paragraph on dispute resolution on request of @Carsten.Schuermann.

[EDIT] Replaced "possible solutions" with "viable or at least potential solutions" after @Olivier.Pereira saw it as overly optimistic.

[EDIT - after dialogue ended] Refined addition after comment by Vanessa Teague (see edit above) to remove a potential ambiguity pointed out by her and other experts via mail after the dialogue ended. Old: "For other experts, establishing internet voting is far less inevitable." New: "For other experts, it is much less clear that Internet voting will earn the trust necessary for being established."

Edited by Christian Folini 4 months ago

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini · 5 months ago Maintainer

Rule number 1: If you write a summary in the middle of the night, sleep over it before you publish it online. I thought it better to give you access to the summary ASAP, but when I read it again today, I found several ambiguities that I have clarified now.

More feedback welcome.

Bryan Ford @Bryan.Ford · 5 months ago Developer

A nitpick: I would suggest changing "during the 15 years" to "during the past 15 years", otherwise this phrase sounds liable to be confused with the reference to the *next* 15 years in the original question.

In the sentence "One politician who claims that a vote has been manipulated can do much more damage than an actual attack", I would change "much more damage" to "as much damage", to avoid stating or implying a judgment one way or another about whether perceived risk or actual risk is more critical, as we've already discussed.

Nitpick: "According to the expert's big picture" -> "According to the experts' big picture" - this just corrects the position of the apostrophe for plural, because there's more than one expert in the dialog.

"to improve the security" -> "to improve voting security" - just a minor clarification

Apart from these nitpicks, excellent summary - looks great to me!

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you @Bryan.Ford for these useful fixes and also thank you for the kind words. I appreciate you still like it since I left many of your ideas out for the summary.

I have adopted all your proposals in the text.

However, the rewording of the damage due to politicians is a bit more tricky. I think your reasoning makes perfect sense. Yet the wording I used was taken from @Oscar.Nierstrasz and I was a bit reluctant to simply change it. But then it's getting late and it is not like the argument is turned around. It's just more balanced or ambiguous.

@Oscar.Nierstrasz: If you really object to this rewording, then please speak up and we can revert.

Other opinions welcome too of course.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

It's fine for me. I'm happy with the change.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Oscar.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

Overall I'm fine with the summary.

Only one small comment: Was it your intention that the paragraph (2) below

(2) "After all, there is no perfectly secure voting system and the voters should have a transparent view on the known weak spots with the voting process. You can not rule out fraud completely nor can you rule out technical errors. But you can make it very hard and expensive to cheat. And you can prepare yourself to be able to detect an attack as soon as possible. By establishing a strong track record, you can build trust."

follows paragraph (1) below

(1) "According to the experts' big picture, the dialog focused on internet voting and left the existing physical voting channels aside. However, the existing voting channels are also prone to attacks; not the least via the auxiliary electronic system components. A holistic view on the possible attack vectors is thus necessary. This should lead to a holistic effort to improve voting security with a risk-based approach."?

If this is the case (e.g. to extend the discussion in the paragarph above to other types of voting), then ok.

Otherwise, I found that (1) breaks the flow as an off-shoot of a more focused discussion and could be moved either to the beginning or right before the summary on the 15 year outlook.

Collapse replies

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you for your comment @Bogdan.Warinschi . The idea is exactly to extend the discussion in the way you describe.

One could argue that the two paragraphs could be melted together, but the resulting paragraph would be rather long, so I'd rather leave it this way.

I understand, that the text is hard to read. Hemingwayapp.com assigns it to 11th grade and states that 8 of 33 sentences are very hard to read (that is down from 15 with my first draft. :)

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

Hello @christian.folini , and all. Again, I'm not sure whether it's necessary for us all to agree, but I don't necessarily share the optimism here:

"If the experts look 15 years ahead, then internet voting is established and the population has confidence that the voting system produces the correct results. For some of the experts, said process will take a bit longer."

Can we please add, "Not everyone agreed that Internet voting would inevitably be established," because I don't.

I agree with everything else that has been said here, about the path forward and the importance of transparency, trust-building, etc. I just think that it's entirely possible that better understanding of the risks will lead to a decision *not* to pursue this direction. I don't think Switzerland has really decided one way or the other - that is the whole purpose of further dialogue, right?

Collapse replies

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 4 months ago Developer

I support Vanessa's comment. I still consider Internet voting as not trustworthy right now and do not trust it. The discussions that we have had these last months are important to move in the right direction. However, I am not sure at all that Internet voting will be trustworthy (and that I will be able to trust it) in 15 years. Moreover, I hope that Switzerland will not promote a large scale Internet voting system as long as it is not trustworthy.

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you for your comment @Vanessa.Teague (and your additional contribution @David-Olivier.Jaquet-Chiffelle).

This statement was missing in the big picture comments above when I wrote the summary.

I have therefore added it now, yet not exactly in the wording you proposed but was follows:

> If the experts look 15 years ahead, then internet voting is established and the population has confidence that the voting system produces the correct results. For some of the experts, said process will take a bit longer. For other experts, establishing internet voting is far less inevitable.

I think is conveys your statement in a similar way as your proposal.

Olivier Pereira @Olivier.Pereira · 4 months ago Developer

Hi, About: "For other experts, establishing internet voting is far less inevitable." Is "inevitable" what people had in mind? My understanding is that we are not trying/willing to avoid it, but rather that we are not sure yet whether, as our understanding improves, there will be an established consensus that Internet voting is a useful and trustworthy voting medium within the set of other options.

Florian Egloff @Florian.Egloff · 4 months ago Developer

Sorry to be late. I agree with @Olivier.Pereira's critique of the language. I certainly do not count myself to the people that see internet voting as established in 15 years (as a prediction). Maybe one could change this to: "For other experts, it is still unclear whether there will be an established consensus that Internet voting is a useful and trustworthy voting medium within the set of other options."

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

And I'm pretty sure that @Olivier.Pereira doesn't either.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

Agreed (thanks @Vanessa.Teague for attracting my attention on that sentence). I think that the problem remains extremely complicated (many of our discussions support this) and full of open questions. I do not know how the answers will look in 15 years, but I strongly suspect that many of the answers that will be found will be valid at one point in time and in one country (or maybe even Canton), and are unlikely to be stable with time, or transferable to other locations.

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

In addition to all that was said, I am missing a sentence about evidence/auditing/dispute resolution, maybe after "By establishing a strong track record, you can build trust.":

"Some experts expressed the necessity for strong dispute-resolution procedure that require the Internet Voting system to produce broadly accepted and immutable evidence that can be independently checked to settle claims of malfunction and cyberattacks."

Collapse replies

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you for bringing this up @Carsten.Schuermann. I left it out on purpose when I wrote the draft summary, since I thought you were the only person mentioning it in your statement. I have now checked again and indeed, @Florian.Egloff is also concerned about this problem.

So I have added your proposal in the position you suggested.

Olivier Pereira @Olivier.Pereira · 5 months ago Developer

I also wonder if the words: "if not most" are not a bit optimistic in "But this dialog has shown possible solutions to many if not most of them."? I guess that this also depends on the meaning of "possible": is it like "this does not sound crazy, and it would be worth exploring whether this solution can work for real" or "we are confident that those three options all solve the problem, we just need more time to clarify some details in order to decide what to do exactly".

Collapse replies

Christian Folini @christian.folini · 4 months ago Maintainer

Thank you for raising this question @Olivier.Pereira. In fact I have added the "if not most of them" after I thought that "many" was a bit naked. I agree the new wording takes it very far, but I had your first meaning of *possible* in mind with my wording. I mean it's *possible* and not *viable*.

Now if you translate this to German, then *possible* is most likely going to end up as *möglich* which implies *viable* to a fairly strong extent. I did not think of that translation when I wrote my text.

If there was time, we could sort this out together, but we need to close this discussion.

So I think it is best to settle on "viable or at least potential solutions". I hope this addresses your concerns and nobody else is offended.

Olivier Pereira @Olivier.Pereira · 4 months ago Developer

I am good with that! Thanks!

David-Olivier Jaquet-Chiffelle @David-Olivier.Jaquet-Chiffelle · 4 months ago Developer

I support Vanessa's comment. I still consider Internet voting as not trustworthy right now and do not trust it. The discussions that we have had these last months are important to move in the right direction. However, I am not sure at all that Internet voting will be trustworthy (and that I will be able to trust it) in 15 years. Moreover, I hope that Switzerland will not promote a large scale Internet voting system as long as it is not trustworthy.

Christian Folini @christian.folini · 4 months ago Maintainer

I have addressed all questions raised throughout the day and I think we have found reasonable solutions for all of them. Let me thus close this conversation.

Many thanks for your valuable contributions and the proof reading of my summary.

Christian Folini @christian.folini removed Last-Call label 4 months ago

Christian Folini @christian.folini closed 4 months ago

Added block 11 and 12
dune73 authored 5 months ago

79fd4a1b

12-future-dialog.md 5.16 KB

Discussion Block 12 - Future Dialog

1. "Future Dialog"

We aim at continuing and structuring the collaboration with independent experts in various domains, e.g. at conceiving, implementing and scrutinizing the systems, risk-, incident- and crisis-management as well as communication. The foundation of a successful collaboration of administrations, providers and independent experts depends on a common understanding achieved by regular exchange.

We believe that the discussions on the platform have contributed to mutual understanding and there would be much more to talk about. The final day of the dialog is approaching. Of course, many of the participants will stay in touch after the dialog and discuss internet voting in other settings and that is a good thing. Nevertheless, we would like to find a reasonable setting that promotes exchange between administration and experts in the future.

2. Related Question

The related questions are labelled Block-12.

Individual links to related questions

- Block 12 Discussion A - The Future Dialog

3. Questionnaire

The thesis is based on the question 5.1 of the questionnaire.

Question	Summary	All Responses Combined	Adamiste Alves Domingues	Basin Capkun	Dubuis Haenni Koenig Locher	Egloff	Ellenberger	Ford	Gilardi	Jaquet-Chiffelle
5.1	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link

4. Download Complete Block and Questions as PDF

Complete Block and Questions as PDF

When editing one of the blocks, please allow up to 1 minute to generate the PDFs anew. The PDFs will not be available during this time and downloads will result in a 404 status code (File not found).

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Discussion 12A - The Future Dialog (Block 12 - The Future Dialog)

Reference to originating discussion block

Block 12 - The Future Dialog

Questions

Would it make sense to run a dialog platform continuously?

Do you see another way of having a continuous dialog?

Or should dialog be rather event driven?

What kind of events could give a good frame to promote exchange on Swiss internet voting?

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini changed due date to December 01, 2021 5 months ago

Christian Folini @christian.folini added Block-12 label 5 months ago

David Basin @David.Basin · 5 months ago Developer

An alternative to a dialog (or complementing it) might be to setup an independent commission with responsibilities to advise, regulate, and possibly even oversee elections. E.g., just like within the UVEK Department in Switzerland there is ComCom (the federal communication commission) there could be an analogous commission within the Bundeskanzlei.

Edited by David Basin 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

In #57 (comment 1111) I raised the possibility of an academic advisory board, quoted here:

> The Federal Chancellery has raised the possibility of an academic advisory board. Based on the discussions in this dialogue, I see merit in such an endeavour. At all times, participants of the dialogue were civil, open, and neutral, and everyone brought their expertise to bear on the relevant issues. For an academic advisory board to be taken seriously, the participants have to be (as was the case in this dialogue) free to publish their contributions to the dialogue. Furthermore, they should explicitly not be deciding but should stay in an advisory capacity. I say this as the election system is a political institution, which means, that changes in such should be decided politically. It is not up to experts to judge which system is the best for Swiss democracy – we can advise, bring good arguments for and against, but not decide. Notably, such an advisory board should not only be a resource for the Federal Government, but also for the Cantons who are considering whether or not, and how, to go forward with regard to internet voting.

In contrast to @David.Basin 's suggestion, I do not see an independent commission to be the best model to *regulate* political processes. I could see merit, though, in the advisory & overseeing functions. I would note, that for the overseeing function, you would not just have to have the expertise, but also political diversity, for the legitimacy of its findings to be accepted.

I think it would be important to draw on this dialogue not just for creating new regulations, but also, to better understand where the boundaries of knowledge currently are. By consequence, one needs to identify which types of academic studies need to be funded, before, during, and after a possible future internet voting trial to address those knowledge gaps.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I guess I already addressed this question as well in my answer to 11A. In particular, I think a dialog such as this could be extremely

beneficial as either a continuous or periodic activity. In addition, a complementary "citizens' advisory board" of some kind comprised of representative non-experts could greatly help with the important usability testing and refinement, education methods and materials, and public trust-building tasks.

Dialog platforms: Using GitLab or a similar platform like this for such a dialog seems perfectly reasonable on a continuing basis, at least for an expert advisory board. I feel that this dialog was extremely well-organized and effective and have no significant complaints. For a citizens' advisory board, other more user-friendly and developer-focused discussion/deliberation platforms like Loomio might be preferable for simplicity and usability reasons. Experimental academic platforms like Deliberatorium and DebateHub may also be worth examining for compelling "dialog process" ideas and mechanisms, even if academic platforms are likely to be less mature or robust.

Events: I think one-time or occasional events could complement, but should not replace, a continuous or regular dialog. Occasional special events might help raise broader public awareness of - and ideally participation in - the progress of and debate about E-voting and voting technologies in general. For example, larger-scale events might usefully be organized around significant new developments or upcoming milestones such as the initial planning and design of a new voting system generation, (much later) the preparations for its deployment, or of course (necessarily but less desirably) an unanticipated crisis point such as the discovery of major flaws or weaknesses.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for the positive feedback about the organization of this dialog.

Out of curiosity: Would a classic mailinglist be equally adequate for such a dialog among subject matter experts, or do you think it takes the advanced functionality of a platform like this one. After all, a mailinglist is *push*, while this platform here is *pull* enriched with spammy *push notifications* via the email channel.

Bryan Ford @Bryan.Ford · 5 months ago Developer

That's a good question indeed. An ordinary mailing list might potentially be adequate, if it is used well, and gives participants the freedom to use their own preferred/familiar E-mail clients. It does have some disadvantages, though.

For example, if participation in the long-term/continuous dialog is rotating (as it probably should be), mailing lists make it harder for newcomers to the dialog to "catch up" on relevant history - yes there are mailing list archives but who really takes the time to dig through mailing list archives? A mailing list service that gives users the convenient option, say, just to "send all the messages from the past 6 months to my inbox when joining the list", or on seeing a new message that's part of an ongoing thread, click a link to "send me all the back-story messages I haven't already gotten from this particular discussion thread", might be useful if such a mailing list manager exists. This ability for newcomers to catch up on history is an oft-cited benefit of messaging platforms like Slack, although I'm personally not a big fan of such platforms either.

The ability to attach quick "likes" or "thumbs-ups" or similar canned responses to messages is also a useful convenience and improvement over the old E-mail/UseNet practice of writing messages like:

```
> [1000-line screed quoted directly]

Agreed!
```

None of these benefits are indispensable, however, that I can tell: e.g., it seems like only a few of us have actually been using the quick-reaction emoji and similar features and only occasionally, so they seem to be features we could live without.

Florian Egloff @Florian.Egloff · 5 months ago Developer

I prefer gitlab over a mailing list as issues can be better separated. I also found the split-up time not the easiest to engage. Perhaps a mix with physical events would be the sweetspot?

Edited by Florian Egloff 5 months ago

Fabrizio Gilardi @Fabrizio.Gilardi · 5 months ago Developer

I've found the current dialog platform well done, but the main problem is the asynchronicity, which makes planning harder. For future discussions, I'd have a strong preference for online meetings. For me, it would be much easier to set aside some time for preparation, participate in the discussion for a couple of hours, and then move on to the next thing. With the current setup, things have been very fluid and that made it difficult for me to engage effectively. I'd rather focus for 2 hours at a time instead of taking ten minutes here and twenty minutes there.

David Basin @David.Basin · 5 months ago Developer

I have a lot of sympathy for what @Fabrizio.Gilardi says. But try to find a time where all of us can meet for a 1/2 day. It won't be easy. And sometimes it is nice to have time to think.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

As the most timezone-challenged participant in the dialogue, I find asynchronicity to be a feature. But I understand it's not a helpful feature for most other participants.

Bryan Ford @Bryan.Ford · 5 months ago Developer

I also actually like the asynchrony of this approach to the dialog. I've had an extremely hectic schedule with lots of conflicting demands lately and having the flexibility to "catch up" on a lot of threads periodically whenever I can has really been valuable, almost essential to my effective participation.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you all for your perspectives on what works and what might not so well. If you allow me to share my perspective:

I see advantages with the focused workshops; ideally on site. You advance quickly that way and you get a lot done in a single day.

The asynchronous written dialog is dragging on. I've in something like a 7x24 mode for several months now and it's really exhausting.

For long stretches of time, you experts (and even more so the other participants) were more reluctant to write something than you would have been if meeting on site. However, the comments and responses we received were of very high quality. You just can't do a workshop with footnotes, but you can write comments with links to publications and I think that is a big value in itself.

@Florian.Egloff proposed a mixed approach. I think that could be beneficial. One could do written discussions and free exchange with less time pressure and then on-site workshops a few times a year to nail things down - or use the workshops as input and inspiration and come to the final conclusions via the written dialog, where people have enough time to think about the various arguments. This would allow to include people who don't feel at ease with a written dialog - and those who just can't make the trip like @Vanessa.Teague.

Either way, nothing is decided and the decision won't be ours anyways. But it is helpful to know where you stand. So please keep the conversation flowing: The platform is still open for new comments for another 27 hours and 37 minutes.

Bryan Ford @Bryan.Ford · 5 months ago Developer

Thanks Christian for your perspective, and I agree that a mixed approach - using occasional time-synchronized (in-person and/or remote) high-bandwidth discussion events to generate ideas, questions, and topics, with more asynchronous text-based discussion in the intervening periods - may be a good balance in the future.

Oscar Nierstrasz @Oscar.Nierstrasz · 5 months ago Developer

I'm not convinced by gitlab as a platform for long-term discussions. It is hard to maintain an overview of the current state of the consensus and of the discussions. I would recommend considering an approach like that of wikipedia, where the current state of the knowledge is highlighted, and the ongoing discussion is in the background. (Maybe it's not so much a question of gitlab itself, but rather the way the information is organized.)

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

The dialogue was not only enjoyable but also particularly informative (personally, i have learned a lot from the well reasoned, thorough and well-articulated contributions of dialogue participants). I see tremendous value in continuing the conversations.

For me the asynchronous format worked quite well (in particular the flexibility it affords in terms of time allocation) and I would favor it over a synchronous event.

I have somewhat mixed feelings about github as discussion platform -- I'm guessing I haven't used its full features but I got only sporadic notifications (which were then classified in somewhat bizarre ways by the google mail client).

Whatever the platform, effective communication via an asynchronous medium would require two elements which worked very well here: identifying the set of questions/topics to discuss and, more importantly, one or more persons to play the mediation role that @christian.folini so diligently played in the current incarnation of the interaction.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for kind words @Bogdan.Warinschi.

As for gitlab as the platform, you are the first one to mention missing notifications.

The various ways to trigger a notification are explained here: https://evote-dialog.okx.ch/christian.folini/evote-dialogue-howto/-/blob/master/HowDoIMakeSureIGetNotifications.md

Are you sure there was a mishap on the platform's side?

During the evaluation, we stacked gitlab against Sharepoint and one of the many online discussion platforms. Gitlab came out as clear winner with an overly complex GUI being one of the drawbacks.

Bogdan Warinschi @Bogdan.Warinschi · 5 months ago Developer

It was probably me (haven't subscribed to all discussions). But when I did get notifications they were classified under various labels by my mailer without any obvious rule that I could deduce (e.g. some came under "Promotions" a folder that I don't check regularly)

Christian Folini @christian.folini · 5 months ago Maintainer

But thank you for the confirmation.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you for your perspectives on a future dialog. Please find my draft summary below.

I got the impression a majority of people in this thread favour the written dialog over face2face meetings. But given this is a written dialog, the numbers are probably skewed, so I left the strengths of the two groups open in the text.

Would it make sense to run a dialog platform continuously?

Do you see another way of having a continuous dialog?

Or should dialog be rather event driven?

What kind of events could give a good frame to promote exchange on Swiss internet voting?

The dialog via the questionnaires and here on the platform is seen as well-organized and effective.

The experts find it very beneficial to continue the conversation in a scientific advisory committee.

The extent of the authority of the committee is not entirely clear: It could have a function with overseeing elections and votes, but it should rather not have regulating power.

Identifying the relevant topics, the preparation of the questions and the moderation of the dialog are seen as key success factors.

Several experts felt uneasy and pressured with the asynchronous written dialog. For other experts it felt beneficial to be able to time the contributions oneself and to think about arguments before posting a response. A mix of written dialog and on-site or remote meetings is likely a balanced compromise.

The use of the gitlab platform for the written dialog was adequate. Yet a lighter approach via a mailinglist could also be an option. Gitlab as well as a mailinglist come with the problem how to organise the current state of the knowlege and the consensus in a dynamic overview. A wiki might be a beneficial replacement or complementary component in this regard.

Next to a scientific committee, there could also be a non-expert citizen's advisory board that supports various tasks like public trust-building, usability testing and refinement as well as education methods and materials.

This is one of the last summaries. It is likely to remain open until the very end. Please give us your feedback, if there none or no negative one, I am going to assume consensus.

Christian Folini @christian.folini added Last-Call label 5 months ago

Christian Folini @christian.folini mentioned in issue #53 (closed) 5 months ago

Christian Folini @christian.folini mentioned in issue #54 (closed) 5 months ago

Christian Folini @christian.folini mentioned in issue #55 (closed) 5 months ago

Christian Folini @christian.folini mentioned in issue #56 (closed) 5 months ago

Carsten Schuermann @Carsten.Schuermann · 5 months ago Developer

Thanks @christian.folini for moderating this. It has been a pleasure to be part of this exercise.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Carsten. It's been an honour to serve this dialogue.

Also thank you for your valuable contributions on many, many fronts.

David Basin @David.Basin · 5 months ago Developer

I like the summary. @christian.folini let me also thank you for moderating. I think you did an excellent job, fairly summarizing all the divergent opinions over this large spectrum of topics, while also challenging us and keeping the group focused in the right direction.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

Many thanks David. Your support with the summaries for the cryptographic questions is much appreciated.

Vanessa Teague @Vanessa.Teague · 5 months ago Developer

Yes, thanks @christian.folini and the other organisers - this has been a very interesting and productive discussion.

Collapse replies

Christian Folini @christian.folini · 5 months ago Maintainer

You are most welcome Vanessa. I probably learnt the most out of this.

Christian Folini @christian.folini · 5 months ago Maintainer

Time to wrap it up. No negative feedback on the summary. I'm thus closing this.

Thank you for the very positive feedback on the dialog, it's been a pleasure.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini removed Last-Call label 5 months ago

Closed Opened 6 months ago by Christian Folini Reopen issue

Discussion 99A - Voting Security Outside of Online Voting

Dealing With The Problem For This Dialog

Several experts have stated that it is too simple to assume Swiss Post is a fully trusted transport channel for the printed voting material. We can not rule out attacks during the transport without looking into the details. So it would be useful to put this on the table for a discussion too.

Still, the focus of this expert dialog is on electronic voting. (a) Because this is actually the mandate given to us, but (b) also because the transport of the printed voting material is but one step where physical voting and electronic voting use the same mechanism. In fact there are many more from working with the electronic voting register to the software aggregating the results per canton (and distributing the seats). So if we talk about physical transport, we should also talk about all the other potential vulnerabilities of voting and if we do that, we lose the focus on electronic voting and (c) I am sure that we will run out of time.

So we can not discuss it as extensively as the core topics of this dialogue. But assuming that it is widely agreed that this is an important topic, we propose a statement to be included in our final report (see below). We invite you to comment on the statement / make alternative proposals.

Previous Work: Killer/Stiller 2019

The potential vulnerabilities of the physical transport and many other potential weaknesses of the voting process are described by *Christian Killer and Burkhard Stiller: The Swiss Postal Voting Process and its System and Security Analysis, Zurich 2019*.

Here is the process for voting by mail as modeled in this paper (the process for voting physically in the voting office would be very close to this one, but a bit simpler):

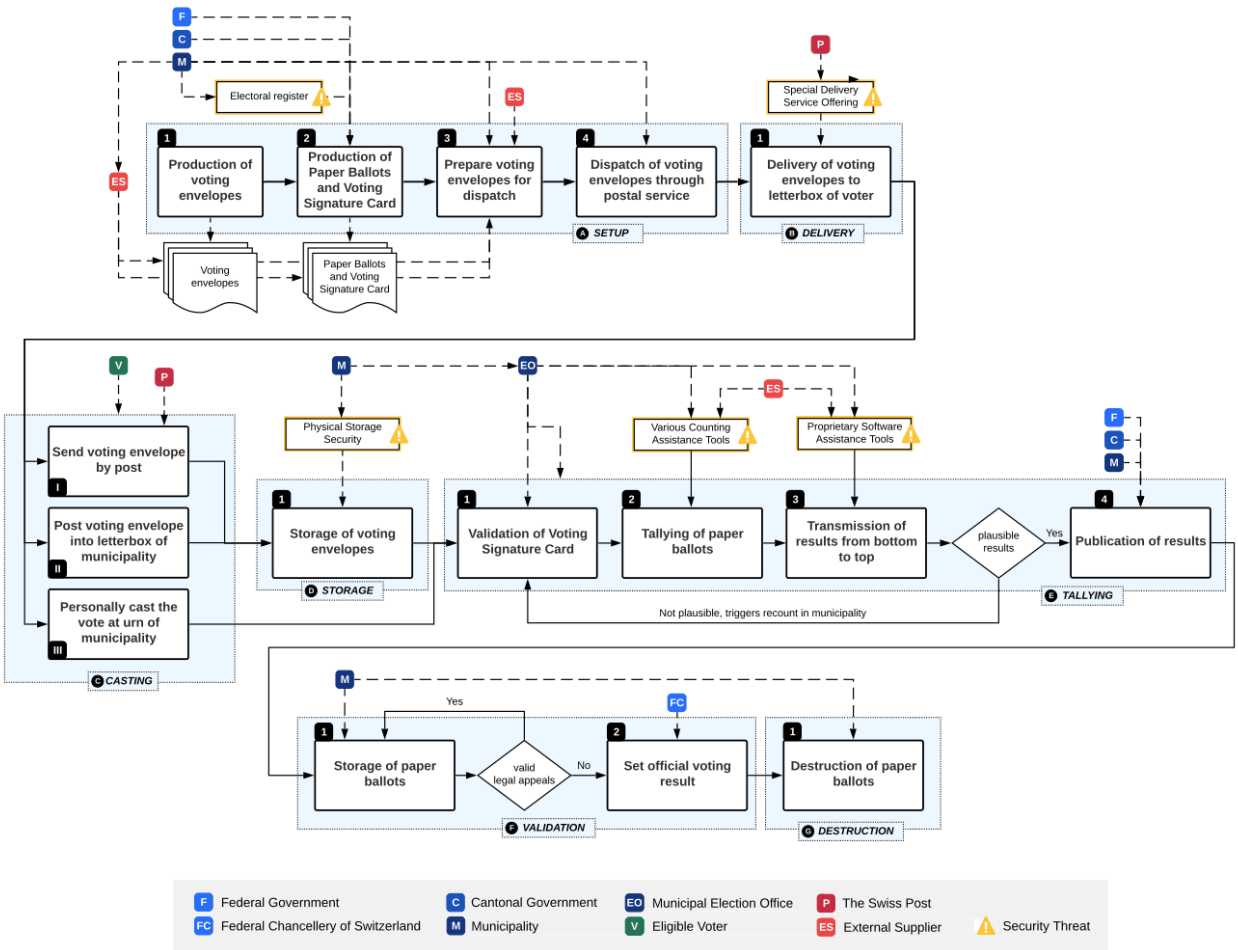


Fig. 3 out of Killer/Stiller 2019

The various administrations are aware of the problems discussed in the paper and they have an earnest interest addressing them.

This is the list of potential weaknesses / threat events brought forward by Killer / Stiller 2019:

- TE1 - Delay production of physical artifacts
- TE2 - ER master records

- TE3 - ER data snapshot
- TE4 - Forge physical artifacts
- TE5 - Steal assembled VEs before dispatch
- TE6 - Re-route VEs
- TE7 - Steal VEs from voter letterboxes
- TE8 - Steal VEs from municipal letterbox
- TE9 - Re-route VEs
- TE10 - Cast stolen or forged VEs
- TE11 - Access stored VEs
- TE12 - Manipulate tallying
- TE13 - Manipulate final tally
- TE14 - Initiate premature destruction

If you see additional threat events or if there is anything else you would like to share, then please do so in the comments below. We will then integrate that into the proposed statement for the final report.

Statement To Include In Final Report

Here is the **statement**, that we will include in the final report of our dialogue:

> The expert dialogue focused on internet voting. Yet online voting is not the only use of electronic systems in the voting process, nor are the electronic systems the only part of the entire voting process that could be attacked successfully. A paper from Christian Killer and Burkhard Stiller ("The Swiss Postal Voting Process and its System and Security Analysis") from 2019 examined the physical voting process and identified several potential weaknesses. We encourage the federal and cantonal authorities to look into these problems in order to remedy the identified vulnerabilities.
>
> However, we also recognize that this paper is only part of the story. For example, Killer & Stiller did not analyze how vulnerabilities in the existing postal voting system could become a threat to election security once internet voting is introduced, and vice versa. These "cross-channel" interaction risks should be examined carefully. Killer & Stiller also did not examine important threat areas that can affect postal and internet voting channels alike, such as voter coercion or vote-buying, a risk that appears most scalable and attractive to foreign adversaries when applied to Internet voting (see "On-Chain Vote Buying and the Rise of Dark DAOs", 2018), but has also proven a realistic threat to postal voting even in mature democracies (see "Election Fraud in North Carolina Leads to New Charges for Republican Operative", 2019).
>
> Finally, the inherent risks of deploying Internet voting must be carefully balanced against the bigger-picture risks of *not* having it, such as the potential disenfranchisement of citizens abroad whose postal services are slow, unreliable, or untrustworthy, or the systemic risk of lock-in to a postal voting technology monoculture that may present few clear paths towards greater long-term security, transparency, or convenience.

[Full transparency: I (Christian Folini) advised Christian Killer on said paper and one or two of the attack scenarios were mine.]

[EDIT] Replaced shorter draft statement with the extended text developed by @Bryan.Ford and @Florian.Egloff below.

Edited 5 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Beyond-EVoting label 6 months ago

Christian Folini @christian.folini mentioned in issue #10 (closed) 6 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

As I raised this in the discussion of 2a, I will go first here.

I consider it my professional responsibility to point to insecurities arising from the introduction of a new voting channel through its interaction with the existing voting system. I would strongly encourage the @federal-chancellery to, if it is not done in this process, to have a separate process assessing these interaction risks.

Existing digital risks in the physical voting process

Killer & Stiller 2019 point out the existing digital risks in the current physical voting process. These are problematic as if the voting channels are exposed to the same threat actors, which in a digitalized physical voting/tallying/transmission of results procedure is the case, one would have to consider the risk profile not only to the internet voting system but also the physical system. A threat actor is likely to target "election security" and not "internet voting security", consequently your risk assessment and mitigations should be working across channels.

Operator of internet voting system & distributor as one entity

Killer & Stiller 2019 do not analyse how the risks would change if the operator of the remote internet vote is the same as the distributor of the remote postal vote, nor do they analyze how the threat profiles would change, if internet voting is introduced. This is problematic and needs to be part of a discussion of the introduction of a new voting channel.

Internet voting is supposed to reassure voters that their vote was cast-as-intended, recorded-as-cast, and counted-as-recorded. If the distributor is controlling both the system and the delivery, then it has an increased potential to undermine the vote. This is important when thinking about single points of failures and cross-over trust problems between channels. (also pointed out as "simplistic" analysis by @Sergio.Alves.Domingues in the answer to the question 2.1.3).

If the physical voting process is supposed to serve as a "backup strategy" for the case of a "failed" internet vote, the independence of the voting channels would be even more important. Thus, one should assess what new problems are introduced, when the distributor of physical mail (in CH the SwissPost) is also managing the online systems, given the significant trust placed into the distributor in the physical remote voting channel.

Is it a problem or strength that the operator is at the same time the distributor of the authentication and return codes? I.e. do we have to "trust" the operator anyway, and hence, we mitigate an additional point of failure when they are unified, or is there an advantage in keeping them independent of one another? My intuition is that for operating the system, the operator does not necessarily need to know who is assigned which codes – or am I wrong in this understanding? This is something that the distributor could potentially learn (by opening the envelopes).

Distributor as "Black Box" in Killer & Stiller 2019

Killer & Stiller 2019 point out that «The Delivery phase B is a black-box» (p.11) and rate the threat event 6 (TE6) as an "availability" problem with "unknown" effort and "unknown" detection. They state:

> "Phase B [delivery] is characterized by the trust placed in one large entity, the SP [Swiss Post]. Thus, the effort and detection probability of TE6 can only be analyzed with additional information or access to internal SP systems, operations, and processes. Generally, however, an insider can achieve a low detection with moderate effort.» (p.12)

I concur but would adapt for our discussion of internet voting: the threat in the physical distribution is not (only) to the availability, but to the confidentiality when internet voting is introduced. Whilst breaking confidentiality of "empty" voting cards is not an issue in the physical remote voting, in internet remote voting this is a compromise of the system. (note on this also @Bryan.Ford 's answer to question 2.1.3.)

Conclusion

It is insufficient to rely on Killer & Stiller 2019 to assess how the existing vulnerabilities in the existing voting system can become a threat to election security once internet voting is introduced (and, though out of scope here, vice versa). Assuming that current mitigations to the current voting system would also mitigate threats to the internet voting channel's use of the current system is underappreciating the new risks that arise through the interaction of the two.

Christian Folini @christian.folini mentioned in issue #28 (closed) 6 months ago

Bryan Ford @Bryan.Ford · 6 months ago Developer

Thanks Christian and Florian. I agree with pretty much everything said so far in this thread.

But while we're on the topic of voting security issues not specific to Internet voting but affecting both channels, here's another major one I'd really like to see called out that wasn't considered in Stiller & Killer: namely vote-buying and coercion. I know I've brought it up before/elsewhere but this seems like a good context in which to raise it again.

To try to summarize briefly, this is a risk that both of Switzerland's current (remote) voting schemes share - postal and Internet - along with current remote voting schemes most anywhere, stemming from the fact that the postal authority can't control the physical environment in which voters cast their votes. This creates numerous types of coercion risks including domestic coercion (dominating family member monitoring how you vote over your shoulder), neighborhood coercion (the village elders will instruct you how to vote, thank you), or outright vote-buying like we saw recently in the L. McCrae Dowless incident in North Carolina. This last is an excellent example that even if we have no reason to believe coercion or vote-buying fraud of this type is widespread in practice, it certainly does happen including in advanced and supposedly mature democracies.

While this risk affects both postal and Internet voting, many of the easily-envisioned coercion or vote-buying attacks may be much more scalable and easy for a foreign adversary to pull off from out of the country with little or no physical presence in Switzerland - and hence with little worry about getting punished if caught. For more on this see for example the article by Juels et al, "Blockchains won't fix internet voting security – and could make it worse", and Daian et al, "On-Chain Vote Buying and the Rise of Dark DAOs".

In many less-stable democracies in the world, it would be unthinkable in good conscious to propose or deploy a (remote) voting system that is completely vulnerable to coercion, because local and regional gangs and strongmen would without question threaten everyone in their neighborhoods or circles of influence with violence or worse if they don't vote the desired way. If Switzerland sets a global example that coercion-vulnerable remote voting is just fine and isn't a problem, then that gives other governments - especially authoritarian governments at all levels - everywhere else in the world the cover, the excuse, and perhaps even reusable open-source voting technology, that they might adopt, saying, "see, it's good enough for Switzerland so it should be good enough for us!" - and suddenly they have a new tool of anti-democratic coercive control over their populations under the thin guise of democracy. Not the ideal Swiss export!

Finally, as I've pointed out before, by making the locally-specialized decision that coercion isn't a problem "here", Switzerland is reducing and limiting its possibilities for international collaboration on [E-]voting systems, reducing the eventual potential international markets of E-voting technologies first developed here. Switzerland is also stuck having to persuade international experts why it's OK to ignore the coercion problem in the Swiss context despite international voting experts almost unanimously considering coercion to be a major issue. This completely "Swiss-specific" approach also further limits the amount of beneficial (to security) scrutiny these E-voting systems receive, and the total amount of funding available globally for the inspection, hardening, and maintenance of these technologies and the many cryptographic and protocol components they build on (e.g., ZK proofs and verifiable shuffles etc as we've discussed elsewhere). I think it's actually very much in Switzerland's self-interest to adopt approaches to remote voting - postal or Internet - that could conceivably be securely applicable internationally as well, including dealing with the coercion problem, even if doing so costs something and that risk is not seen as of critical priority purely internally.

Bryan Ford @Bryan.Ford · 6 months ago Developer

Further building on the above comments about (a) the need to consider carefully the risks embodied in the currently-dominant postal voting system, and (b) the need to consider unaddressed risks such as coercion or vote-buying that affect both channels, I'd like add a comment about what I feel these risks mean for the E-voting program as a whole. I know this probably steps further outside the formal scope of the current discussion's mandate, but I feel it's a point I need to make.

Balancing long-term systemic risks against short-term technical risks

In the rest of this dialog we're identifying a lot of important risk factors in E-voting and potential ways those risks may be addressed. But counterbalancing all the risks involved in the deployment and use of either the current-generation or a future-generation Swiss E-voting system are the important systemic risks in *not* doing so. Today's reality is that Switzerland is currently almost completely dependent on a postal-only remote voting system, which embodies plenty of under-appreciated technological flaws and risks of its own as discussed above. And it seems unlikely in the extreme that most Swiss voters or cantons would be willing to give up the convenience of remote voting to return to the in-person voting processes predominant in most other countries.

E-voting has its risks, but also holds many possibilities for mitigation of those risks through improvements like the many we're discussing here. Postal voting has its risks too, but beyond that, represents a fundamentally non-transparent dead-end voting technology. I see no feasible way to "fix" the main security, privacy, or transparency weaknesses in postal voting in either the short or long term.

So given the reality that Switzerland is in a sense "addicted" to the convenience of remote voting, but the currently-dominant postal-voting system presents important risks with no reasonable prospect of solutions, I personally feel that by far the biggest systemic risk to Switzerland's voting processes is to get "stuck" with postal voting with no near-term, concrete and well-funded path towards something better. From a big-picture perspective and in the long term, I see this risk of getting stuck with postal voting alone for many years as a much worse, fundamental systemic risk than most of the specific technical risks that are not yet adequately addressed in the current-generation E-voting design. Postal voting is not "good enough" and needs a replacement. No such replacement will be quick, easy, cheap, or risk-free, but it is necessary.

In particular, while in other discussion threads I have been arguing for and will continue to argue for many improvements to the E-voting design - e.g., diversity in control component implementations and verification tools, transparency via some form of public bulletin board, etc. - nevertheless I personally feel that most of these significant design improvements can and perhaps most likely should wait for the next "major version" upgrade to the overall Swiss E-voting design. I personally feel that the current-generation design can and should be approved and placed into service for use by limited voter populations (e.g., those with particular reasons to need it, such as expats) - provided of course the known implementation flaws can be adequately addressed and the implementation thoroughly validated. Even if use is limited to a small population, the existence and deployment of the E-voting system as a partially-independent alternative to postal voting provides at least some valuable diversity in the overall voting system, serves real needs of many voters, and just as importantly preserves the momentum of a positive development path towards better long-term solutions.

In summary, I see the risk of losing that momentum and adequately-funded development towards better solutions, beyond postal voting, represents a much greater, long-term systemic risk to Swiss democracy than the short-term risks of deploying and using a decent but not-yet-perfect E-voting system among limited voter populations. I know that other experts are likely to disagree with me on this; I invite them to comment as they like. But I want to make my position on this over-arching "balance of risks" clear, especially as I continue to point out risks and propose (likely painful and expensive) potential E-voting design enhancements in other parts of this discussion.

Bryan Ford @Bryan.Ford mentioned in issue #27 (closed) 6 months ago

Christian Folini @christian.folini mentioned in issue #33 (closed) 6 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

@christian.folini I have taken a crack at adding to your draft statement. @Bryan.Ford: please do add to it / change it as you think is needed.

The expert dialogue focused on internet voting. Yet online voting is not the only use of electronic systems in the voting process, nor are the electronic systems the only part of the entire voting process that could be attacked successfully. A paper from Christian Killer and Burkhard Stiller ("The Swiss Postal Voting Process and its System and Security Analysis") from 2019 has taken a close look at the physical voting process and identified several potential weaknesses. We endorse the federal and cantonal authorities to look into these problems in order to remedy the vulnerabilities described by this paper. However, we also recognize that Killer&Stiller 2019 did not analyse how the existing vulnerabilities in the existing voting system can become a threat to election security once internet voting is introduced and vice versa. Such an analysis would be necessary, should internet voting be introduced.

Bryan Ford @Bryan.Ford mentioned in issue #32 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #24 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #57 (closed) 5 months ago

Florian Egloff @Florian.Egloff mentioned in issue #20 (closed) 5 months ago

Bryan Ford @Bryan.Ford · 5 months ago Developer

Thanks @Florian.Egloff for your work on the paragraph. Here's the result of my own pass... Is this too long?

> The expert dialogue focused on internet voting. Yet online voting is not the only use of electronic systems in the voting process, nor are the electronic systems the only part of the entire voting process that could be attacked successfully. A paper from Christian Killer and Burkhard Stiller ("The Swiss Postal Voting Process and its System and Security Analysis") from 2019 examined the physical voting process and identified several potential weaknesses. We encourage the federal and cantonal authorities to look into these problems in order to remedy the identified vulnerabilities.
>
> However, we also recognize that this paper is only part of the story. For example, Killer & Stiller did not analyze how vulnerabilities in the existing postal voting system could become a threat to election security once internet voting is introduced, and vice versa. These "cross-channel" interaction risks should be examined carefully. Killer & Stiller also did not examine important threat areas that can affect postal and internet voting channels alike, such as voter coercion or vote-buying, a risk that appears most scalable and attractive to foreign adversaries when applied to Internet voting (see "On-Chain Vote Buying and the Rise of Dark DAOs", 2018), but has also proven a realistic threat to postal voting even in mature democracies (see "Election Fraud in North Carolina Leads to New Charges for Republican Operative", 2019).
>
> Finally, the inherent risks of deploying Internet voting must be carefully balanced against the bigger-picture risks of *not* having it, such as such as the potential disenfranchisement of citizens abroad whose postal services are slow, unreliable, or untrustworthy, or the systemic risk of lock-in to a postal voting technology monoculture that may present few clear paths towards greater long-term security, transparency, or convenience.

Florian Egloff @Florian.Egloff · 5 months ago Developer

Reads well to me!

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you guys for refining my draft and adding important aspects to the discussion and the summary. I am adopting it and will now close this discussion.

Christian Folini @christian.folini closed 5 months ago

Christian Folini @christian.folini reopened 5 months ago

Christian Folini @christian.folini changed the description 5 months ago

Christian Folini @christian.folini closed 5 months ago

Closed Opened 6 months ago by **Denis Morel** Reopen issue

Information about situation of Scytl source code and Swiss Post

Dear experts,

With this message I want to inform you officially about a change in the situation by Swiss Post. Swiss Post decided last year to take over the code from Scytl and to continue itself the development and offer a system which is entirely developed in Switzerland. We signed the contract during April. The main reasons are the proximity of the development to the Swiss market and Swiss Post's strategy for digital services.

Regards

Denis

Drop or upload designs to attach

Linked issues **0**

Denis Morel @Denis.Morel added Meta label 6 months ago

Christian Folini @christian.folini · 6 months ago Maintainer

Above is a message from the Swiss Post Online Voting project lead. It clears up some newspaper articles that appeared in Switzerland during the weekend - and a whole lot of tweets of course.

Swiss Post is not yet ready to answer a lot of questions about this and this is not really a Swiss Post forum, but if you have a pressing question, then give it a shot.

CC @experts

Adrian Perrig @Adrian.Perrig · 6 months ago Developer

It would be great to hear more details on this, especially what their approach is to achieve high assurance code. How many FTEs are working on the project? What is the time line? What is the long-term plan to support this system?

Christian Folini @christian.folini · 6 months ago Maintainer

Swiss Post is currently preparing proper information for the public. I have it on good notice, that several of your questions will be answered in said message.

FedCh has asked Denis Morel to provide the experts in this dialogue with some information ASAP to make sure everybody here is on the same page. You could say, you got a sneak preview.

[EDIT: Reworded first sentence]

Edited by Christian Folini 6 months ago

Denis Morel @Denis.Morel · 6 months ago Developer

@Adrian.Perrig I can give you more information regarding your question. We communicated publicly last week.

We built up a development team with senior profiles from different disciplines (Java-Developer, JS-Developer, Security-Specialist, Architect, Crypto-Developer, …). We worked with Scrum with sprint of two weeks. We decided to have two Product Owners (a Crytpo-Specialist and an operation-Specialist). The decision was taken so, because operation is a very important part of the quality and security of the platform and it is important that operations are near to development. We plan to have 10 persons active.

The quality will be ensured through various measures (peer reviews, pull requests with 4-eyes principles, external reviews, …). It is important also to have an effective continuous improvement process, also for development processes.

Since middle of April we are working ourselves in the code. We set a plan to address different elements, in particular the auditability of the code. The plan is a result of a deep software due diligence. The plan of the publication will be defined together with the

cantons and the Federal Chancellery. The results of the dialog can have an impact on the plan.

The decision of Swiss Post is a long term decision, based on a deep analysis, and it is anchored in the new strategy of the company.

Hope that it answers part of your question.

Adrian Perrig @Adrian.Perrig · 6 months ago

Developer

Thanks for the additional information!

Christian Folini @christian.folini closed 5 months ago

Closed Opened 7 months ago by **Christian Folini** Reopen issue

Status Report: 2020-05-10

Dear all,

Last Monday, we launched the written dialogue around Swiss Internet Voting on this platform here. This is an informal status message looking back at the week:

We invited 25 experts to join the discussion. Another 30 people are participating from various government agencies on federal and cantonal level, as well as several people representing the system provider Swiss Post.

Most of the experts have edited their profile and almost all of them have activated their account.

We got a bit of a slow start which we anticipated: This is namely because of the impressive amount of text that lays at the base of this written discourse: 40 pages of summary based on over 200 pages of questionnaires. It's a lot to read.

Several experts told us that they need a bit more time to read through all the documents and make themselves familiar with the positions of the other participants. Other professors are still very busy with teaching or finishing some papers. Besides, we are aware that the launch of the written dialogue came a bit on short notice.

But whatever, the discussions are getting more lively now - namely tonight - and I am expecting many more contributions during the week. So far, we had 27 comments / answers in the dialogue.

It seems that the Thesis (Discussion A) about the Cryptographic Effectiveness (Discussion Block 1) is finding approval. Even if there are 1-2 things that might have to be clarified. I will try to sort this out quickly, so we can at least close this discussion. The other questions in this first discussion block will be left open for a few more days, likely the whole week.

The next discussion block arranged around the topic of verifiability is almost ready to be launched. This is possibly happening on Tuesday or Wednesday.

If you have any questions or feedback on the dialogue, please comment here or use the general feedback thread.

With this I close this status message and wish everybody a good week!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta label 7 months ago

Christian Folini @christian.folini added Status label 7 months ago

Christian Folini @christian.folini removed Status label 7 months ago

Christian Folini @christian.folini added Status label 7 months ago

Didier Steiner @Didier.Steiner closed 7 months ago

Christian Folini @christian.folini reopened 7 months ago

Christian Folini @christian.folini closed 6 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Status Report: 2020-05-17

Dear all,

We have seen a very busy week with over sixty comments on the platform and we stand at 93 comments now. Surprisingly, the comments did not focus on the new discussion block about the desired level of independence of the control components and the verifier, but on a heated discussion about the power of state sponsored actors in discussion block 1, question F.

In fact I promised to try and have the initial thesis of block 1 updated in the previous status, but 1F became so fundamental, that we think it is better to integrate the result of this discussion back into the thesis. The point with 1F is that if we assume secret services to be able to break any cryptography at will and namely those schemes used in online voting, then the whole endeavor becomes futile. The other side of the medal is of course, that if they have these capabilities, would not it be easier for them to break / undermine social networks and control voting results that way?

As mentioned before, the block 2 is not seeing much attention. This may have to do with @Bryan.Ford giving a very thorough coverage in 2A as to the why multiple different control components are so important. However, I think it is still worth to share your thoughts or at least support Bryan's statement. Because the louder and clearer the opinion of the experts is voiced, the harder will it be to brush it a way.

And if you think Bryan is not right, then it is even more important you resist his opinion, because so far it looks like independence and heterogenity of the control components should be pushed to the very limit which could mean serious operational risks.

I have closed the first two discussions. 1B (-> Block-1 Question B) and 1D. The procedure I used was as follows: I would try and do a summary and ask the participants in the discussion to vet my summary. At the same time, I would label the discussion as "Last Call" and also put in that status in the overview page (README). Then there was feedback that made me re-adjust my summary in one case and then I waited another two days until I closed the discussion. So closing a question is a multi-step process and you have enough time to bring in additional arguments before a question is closed. I plan to use the same procedure for the next discussions we are going to close.

Looking forward to this week, we are going to launch the third discussion block about the printing office. A lot of the experts thought the printing office to be a weak spot in the defense architecture. So there is going to be a proposal on how to improve that.

I wish you a good week!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

[EDIT: several typos and a link to the README]

Edited 6 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 6 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

@christian.folini: Thank you for this. just a procedural point of feedback: Could you please make sure that the days you wait after labelling the discussion "last call" are working days? As for myself, I do not work on the weekends, and hence, I would not "count" these days. Also, I would make it 4 working days, as most of us are also working on other projects and will not be able to follow the discussion every day. BTW: it may not have mattered in this instance, but in future instances, it may. Thank you, Florian

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you Florian. I see the problem here.

So far, all participants and everybody on the notification list of a discussion get a notice about the plan to wrap it up. That's automatic.

But it is true that people who have not yet participated in a discussion and don't check the platform like every 2 days miss the moment to place their arguments. And we also do not want to spam everybody with notifications they need to return to the platform

asap.

If you want to be sure this does not happen to you in any discussion subscribe to the "Last Call" label. This is explained as variant 4 in the Notification How-To.

Committing to a 4 working days grace period would probably work now, it might be a killer towards the end of the dialogue. So let's say I will try to keep it open for 4 working days when possible, but if you really want to be sure, then make sure you get notifications from questions that you are interested in.

Florian Egloff @Florian.Egloff · 6 months ago

Developer

Thank you. I will certainly subscribe!

Christian Folini @christian.folini changed the description 6 months ago

Christian Folini @christian.folini closed 6 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Status Report: 2020-05-24

Dear all,

This is the 3rd informal status report of our experts dialogue.

You wrote over 48 comments to the open questions last week and we stand at 141 comments now. We're about to close discussion 1E about the formal methods and I will try and sum up 1C tomorrow.

The block 2 about the independence of the control components saw a lot of good comments with almost all of them pointing in the direction of more independence. Thank you for contributing. We'll assess the situation during the week on that front and might then propose summaries or get back to you.

We launched the 3rd block with two propositions to secure the print office right before Ascension. This block has not seen any responses so far. I understand it is tricky to make the first step here and make a judgement on a new procedure. But I invite you do give it a go nevertheless.

The next discussion block will focus on public bulletin boards and we plan to launch it during the week.

On some other news, @Matthias.Stuermer has received his Venia Docendi from the University of Bern last Week. Congratulations!

And finally the news is out about our expert dialogue: The NZZ covered it in an article about Swiss Post and Scytl and named @Vanessa.Teague, @Olivier.Pereira together with BFH and especially @Rolf.Haenni as participants.

Have a good week!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 6 months ago

Christian Folini @christian.folini changed the description 6 months ago

Christian Folini @christian.folini closed 6 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Status Report: 2020-05-31

Dear all,

This is the 4th informal status report of our experts dialogue.

We wrote around 50 comments to the open questions last week and we stand at 193 comments now. I just closed 1C (abstraction level of specifications) and the complete discussion block 2 about soft- and hardware diversity is coming to an end with the a summary for 2A-D that has received positive feedback so far. I plan to close this discussion on Tuesday or Wednesday.

Talking about the print office in block 3 did not quite work as planned. We saw several questions on the platform with the 2 propositions and there were also phone calls to clear out some misunderstandings. So we will work on block 3 and get back to you when it is ready to be picked up again.

Meanwhile we launched the discussion block 4 on public bulletin boards. The Swiss internet voting systems do not come with a public bulletin board so far, yet several experts brought up this topic in their responses to the questionnaire. We would therefore like to hear your opinion about the idea in general and about two concrete variants in particular.

There is also a bit of a fringe discussion happening in 99A - Voting Security Outside of Online Voting. The question came up in another discussion and we do not want to make it a discussion block of its own. But we are proposing a statement to be included into the final report. Florian Egloff has already contributed additional thoughts why this is so important and we'll keep this open for some additional time to shape the statement accordingly and give it the weight it deserves.

I was in touch with several experts last week to better understand their perspective on the dialogue. The feedback was generally positive, which we appreciate very much of course. The written form of the dialogue comes with a few downsides though. People are reluctant to make quick statements. Instead they want to think an argument really to the end, before they write it down. This takes time of course, as does the asynchronous discussion mode that forces people to zoom into a discussion and then do this again a couple of hours or days later when a reply or new comments comes in. This is cumbersome. People are also reluctant to contribute in a topic, where they do not feel really, really competent (unlike an oral discussion where people are quicker to express a gut feeling). And finally, writing a comment without new arguments - a simple "I agree" message - feels a bit bland, so there is a lot of silent approval (or maybe also silent disagreement from people that lack the time or the scientific arguments to oppose a statement).

As a consequence, the dialogue has a relatively slow pace, but it comes with a very good quality. So that's just the way it is and I am the last one to complain about quality and the civilized tone in all the discussions (thanks for that btw).

What we can do, though, is using the Thumbs Up (and Thumbs Down) emojis to express approval (or disapproval). This is quick, it signals an opinion and - that is noteworthy too - it does not result in notifications to the other contributors in a thread.

This means, when you upvote a comment, everybody sees that comment and its arguments are finding approval, but you are not spamming people with "me too" notifications. Seeing that more experts agree with a statement is fairly important for me as a moderator as well, as I can then give such an opinion more weight in summary.

So I invite you to make use of this functionality next week. Maybe you want to try it out immediately with this status message. :)

I wish you all a good week and look forward to your contributions!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Edited 6 months ago by Christian Folini

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 6 months ago

Christian Folini @christian.folini changed the description 6 months ago

Christian Folini @christian.folini closed 6 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Status Report: 2020-06-07

Dear all,

This is the 5th informal status report of our experts dialogue.

It seems that we typically get around 50 comments per week on the various discussion. This time it's been 56 and we stand at 249 now. The *Last Call* on the diversity block in 2A triggered an intense discussion, several rewordings and the addition of important keywords. The discussion block is still open, but unless there is additional feedback, it's going to be closed Tuesday or Wednesday.

The structural problem with securing the printing office in block 3 got more input and a noteworthy contribution by @Bryan.Ford that might indeed solve the problem (admittedly, for a very high financial price).

The discussion around a public bulletin board is in full swing now. We received comments on all the eight questions, that we asked. I am sure this is going to continue throughout the week. Ideally, the discussions will also expand beyond cryptography here. We would - among others - value the opinion of political science a lot.

Other than that, we will be launching a new discussion on *Mandated Examinations* during the week and we are currently working on a schedule for the remainder of the dialog. There are going to be half a dozen additional discussion blocks, likely at a bit a faster pace. We plan to publish this plan during the week, certainly on the overview page, possibly also with a separate announcement.

Have a good week everyone and please keep those comments coming!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 6 months ago

Christian Folini @christian.folini closed 6 months ago

Closed Opened 6 months ago by **Christian Folini** Reopen issue

Status Report: 2020-06-14

Dear all,

This is the 6th informal status report of our experts dialogue.

The week got a bit of a slow start, but then things really started to rake up: We stand at 292 comments on the platform now (43 since the last report). The block on the Mandated Examinations has been launched and it saw the first comments today (thank you @David.Basin).

Since we finished block 2 (diversity), I will now move and start to round up block 3 (print office) and later in the week block 4 (Public Bulletin Board). And in case you are interested, the separate discussion on security beyond internet voting got a bit of a boost last week as well.

As previously mentioned, we have updated our planning for the remainder of the dialog. Here are the discussion blogs that we are currently preparing:

Block	Title
6	Development and Publication
7	Public Intrusion Test / Bug Bounty
8	Risk Management and Individual Risks
9	Risk Limiting Audits and Plausibility Checks
10	Forensic Readiness
11	The Big Picture
12	Future Dialogue

We are also trying accelerating the pace a bit. The idea is to launch two blocks per week. For the coming days, I'm quite confident with block 7, but block 6 is a tricky one, so don't hold your breath.

Have a good week!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 6 months ago

Christian Folini @christian.folini closed 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Status Report: 2020-06-21

Dear all,

This is the 7th informal status report of our experts dialogue.

The week brought in another 40 comments and we stand at 332 comments. As anticipated, block 6 on development and publication of the source code is a bit harder to prepare. So give us another day or two until we are ready to launch it. But we started block 7 on the Public Intrusion Test / Bug Bounty. Here, we want discuss a possible evolution of the previous PIT and the idea to make it into a permanent program with a wider scope. If you agree with this, or if you have completely different ideas, please join these discussions.

Meanwhile, I'm rounding up block 3 on the printing office and block 4 on public boards now. Several discussions in block 4 are now marked with Last Call and the other ones will follow shortly. My summaries for block 4 have not received any feedback (and I'm not entirely sure this is a good sign...).

Meanwhile block 5 on the mandated examinations got quite a bit of feedback namely on the fact that it seems to limit the scope of these examinations. We hope this discussion will continue this week since there is more to say there (and more questions to explore).

I wish you all a good week and look forward to your contributions!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Linked issues **0**

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Status Report: 2020-06-28

Dear all,

This is the 8th informal status report of our experts dialogue.

The week brought in a whopping 125 comments. I did not believe the numbers at first, but then I double-checked them and I can confirm: we stand at 457 comments. So we have accelerated the pace as I had hoped. More comments is a good thing, since it's only two weeks to go now.

Last week saw the launch of three blocks:

- Block 6: Development and publication
- Block 8: Risk management and individual risks
- Block 9: Risk Limiting Audits and plausibility checks

Block 6 got a good start. The question of the development model has attracted almost a dozen comments and I am sure we will hear more opinions there. But also the other three questions in the block got some attention. Block 8 and 9 have not been addressed so far, but I am confident that will change soon.

Block 3 - the printing office - has been closed. In block 4 - the public board - we have closed the first discussions, the other ones are still in progress and I will try and round up block 5 (mandated examinations) during the week. Block 7 on the PIT / Bug Bounty Programs is also coming along nicely in my eyes.

Block 10 on Forensic Readiness is - sorry for the pun - almost ready and will be launched shortly. And then we'll do a final Big Picture block that will also include one or more questions on how to continue this dialogue in the future.

Have a good week, stay healthy and please keep the many, many interesting contributions coming!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Edited 5 months ago

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 5 months ago

Christian Folini @christian.folini changed title from **Status Report: 2020-06-21** to **Status Report: 2020-06-28** 5 months ago

Christian Folini @christian.folini closed 5 months ago

Closed Opened 5 months ago by Christian Folini Reopen issue

Status Report: 2020-07-05

Dear all,

This is the 9th informal status report of our experts dialogue.

The conversations grew even more intense last week. A full 182 comments were written on the platform, which brings us to 639 in total. And given this is meant to be the final week of the dialog, I would not be surprised if we would match this in the next few days. In retrospect, those last ten days felt the best for me. Very active discussions, arguments in multiple directions and and many, many interactions in discussion threads.

Looking over the discussion blocks, this is how I see their status:

Title	Status
Discussion Block 1 - Cryptographic Effectiveness	Mostly done
Discussion Block 2 - Diversity to support security and trust-building	done
Discussion Block 3 - Print Office	done
Discussion Block 4 - Public Bulletin Board	Mostly done
Discussion Block 5 - Examinations Mandated by Government	First summaries written
Discussion Block 6 - Development and Publication	Broad discussion, not entirely comprehensive from my point of view
Discussion Block 7 - Public Intrusion Test / Bug Bounty	Broad discussion, comprehensive, but I need to think it through again
Discussion Block 8 - Risk Management and Individual Risks	Some comments across the board, but not very comprehensive so far
Discussion Block 9 - Risk Limiting Audits and Plausibility Checks	Only a single comment
Discussion Block 10 - Forensic Readiness	Very few comments
Discussion Block 11 - The Big Picture	Very few comments
Discussion Block 12 - Future Dialogue	Ideas and Feedback welcome

The blocks 8 and especially 9 and 10 would profit from your thought a lot, but the Big Picture in block 11 should not be forgotten either, of course.

I will start the week with writing summary responses for the questions where I think we are ready. We still "owe" you an updated thesis for block 1 and with block 4 it is difficult to really bring all the things that have been said together.

When I think there is some contradiction or certain aspects have not been considered, I might post followup questions and invite you by name to add your thoughts (-> Notification message). Together, we'll get there.

The Federal Chancellery has told you, that you will be able to adjust the responses to the questionnaire before they will be published. If you want to make such adjustments to your questionnaire, then please get in touch with @oliver.spycher / @aurore.borer now. No immediate plans to publish, but the papers are being prepared now.

It's been almost two weeks, since the media release about our dialog was published. Many news outlets covered our dialogue. In case you do not live in Switzerland pr you do not follow the news, here is an overview:

General new sites

- https://www.tagesanzeiger.ch/bund-ueberarbeitet-grundlagen-fuer-e-voting-versuch-288163399826
- https://www.nau.ch/politik/bundeshaus/e-voting-bund-bespricht-neuausrichtung-mit-experten-65729774
- https://www.swissinfo.ch/ger/schweiz-demokratie-abstimmung-e-voting-moratorium-volksinitiative-zurueckgezogen/45855362
- https://www.swissinfo.ch/fre/suisses-de-l-%C3%A9tranger_le-vote-%C3%A9lectronique-d%C3%A9barrass%C3%A9-de-son-%C3%A9p%C3%A9e-de-damocl%C3%A8s/45855920
- https://www.mittellaendische.ch/2020/06/28/e-voting-dialog-mit-der-wissenschaft-zur-neuausrichtung-des-versuchsbetriebs/
- https://www.bazonline.ch/bund-ueberarbeitet-grundlagen-fuer-e-voting-versuch-288163399826 (SDA basiert)

ICT news

- https://www.netzwoche.ch/news/2020-06-24/die-schweiz-nimmt-den-naechsten-anlauf-fuers-e-voting
- https://www.ictjournal.ch/news/2020-06-24/la-confederation-prepare-de-nouveaux-essais-pour-le-vote-electronique (based on netzwoche.ch)
- https://www.inside-it.ch/de/post/bund-und-kantone-wollen-e-voting-neu-ausrichten-20200623
- https://www.i-web.ch/de/aktuelles/neuigkeiten/welcome.php?action=showinfo&info_id=55417

And now I wish you a good start of the week and hope to read you soon!

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 5 months ago

Christian Folini @christian.folini closed 5 months ago

Closed Opened 5 months ago by **Christian Folini** Reopen issue

Status Report: 2020-07-12

Dear all,

This is the 10th informal status report of our experts dialogue.

The normal exchange stopped on Friday and we have started to round up the remaining summaries.

Last week brought in another 98 comments and we currently stand at 737. It makes me happy to see that we have responses to the Big Picture Question from 11 different experts.

22 questions are closed with a summary. With 17 additional ones, there is a draft summary and I am going to write the remaining 7 summaries tomorrow.

So we're on good tracks. I am confident we can do a balanced summary in the remaining questions. There are a few discussions, where we are not quite there yet: 4A might take some refinement, the Public Bulletin Board was a very hot topic after all. In 6B, it is not quite clear how strongly an Open Source license should be recommended or required and in 8E we might need another iteration to find the common ground of the experts involved.

I'm looking forward to the final days in this dialog. The idea is to close all discussions with a confirmed summary for each on Wednesday.

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 5 months ago

Christian Folini @christian.folini closed 4 months ago

Open Opened 4 months ago by **Christian Folini** Close issue

Status Report: 2020-07-17

Dear all,

This is the 11th informal status report of our experts dialogue and also the final one.

All summaries have been finalized and all the questions have been closed. The dialogue here on the platform is over.

Since Sunday night, we wrote another 123 comments and we close at 860 comments. That's a mean of 80 comments per week for the dialogue that lasted for almost 11 weeks. I'm impressed.

It's been an immense pleasure and a very big honour to serve this dialogue and I thank you for your constructive contributions and the support you gave me and my role.

The Federal Chancellery will be in touch in the next few days with new information.

I tried to freeze all issues so they could no longer be edited, but GitLab does not allow this (unless I am kicking you off the project). So we count on you to leave everything as is. If you miss the dialog already in the morning, then I invite you to leave a comment in the feedback thread that is still open.

Also, if you want to retrieve any of your statements for future reference, please do so. The platform here will remain open for at least a couple of weeks.

All the summaries will make it into a report that is going to be published.

Thank you and goodbye.

CC: @experts, @fedch-extended, @federal-offices, @uag-cantons

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta Status labels 4 months ago

Open Opened 7 months ago by **Christian Folini** Close issue

Meta Discussion: Feedback and Questions about the dialogue and or the platform

- If you have feedback - good or bad - on the platform, please place it here.
- Is there something that is not properly documented? This is the place to ask.
- Anything amiss? Something is not working as it should? Please indicate it here.

You can also reach me directly via email (christian.folini@netnea.com) or twitter @ChrFolini.

Drop or upload designs to attach

Linked issues **0**

Christian Folini @christian.folini added Meta label 7 months ago

Christian Folini @christian.folini mentioned in issue #9 (closed) 7 months ago

Florian Egloff @Florian.Egloff · 6 months ago Developer

@christian.folini : Is there anywhere that we can get an overview of how many blocks there are and what they cover so that we contribute the answers to the block most adequate for that discussion? 5 are listed in https://evote-dialog.okx.ch/federal-chancellery/evote-experts-dialogue/-/tree/master but that probably is not an exhaustive list, right?

Christian Folini @christian.folini · 6 months ago Maintainer

We talked about this in our telco today. The plan was to build such an overview as we move along. After the first two weeks, we get the feeling that we are moving relatively slow, so we will have to set priorities, take responses from the questionnaire without further discussing them and maybe also skip certain questions.

So we are now setting up a rough plan that will then be published at the position you indicated.

Vanessa Teague @Vanessa.Teague · 6 months ago Developer

I like the platform and I think it works well. The list of discussion-pointers on the master readme is very helpful. Any more specific pointers to a particular issue that you wanted us to discuss would help too, but the whole thing is working well for me.

I find the level of discussion pretty good - in many cases, we mostly agree, and it helps that there isn't too much discussion because it's possible to log in and read through a moderate number of responses. Too much more discussion could get unwieldy and make it impossible for everyone to read what everyone else had said.

Collapse replies

Christian Folini @christian.folini · 6 months ago Maintainer

Thank you very much Vanessa. I'm very happy to hear that it works for you and our efforts have not been futile.

We got a several feedback messages outside of the platform as well, and will probably try some optimization.

Florian Egloff @Florian.Egloff · 5 months ago Developer

Hi Christian On the 17. June you announced you aim to close the discussion on all blocks on the 10. July. I will be on holiday from 4. July-17. July and will not answer any questions at that time. Might there be the possibility to make sure all the blocks are either available beforehand or that I could comment after that period? Particularly, I do not think that we have had a discussion on the

wider state of internet voting technology (Big Picture), nor are there any questions re the process of how one would go about introducing such technology. Thank you very much, Best wishes, Florian

Christian Folini @christian.folini · 5 months ago Maintainer

Hi Florian, good to plan the holidays a bit.

The overview over the discussion blocks is at https://evote-dialog.okx.ch/federal-chancellery/evote-experts-dialogue/-/blob/master/README.md

The following blocks have not been launched so far:

- Block 8 - Risk Management and Individual Risks
- Block 10 - Forensic Readiness
- Block 11 - The Big Picture
- Block 12 - Future Dialog

It is very likely, block 8 is being launched on Friday. Preparations mostly done.

Block 10 is making good progress, let's say early next week.

Block 11 and 12 are likely going to be melted together into a single block; ideally published next Wednesday.

Aside, we must not forget, that block 1 needs to be revisited as well.

So considering your holiday plans, I am confident you will have a chance to contribute to all the questions in time, but you won't be around when the summary is being discussed.

The idea is to close the dialog on July 10, so you should reserve a bit of time before you leave. Submitting something afterwards is probably too late, since it would not make it into the summary in time.

Florian Egloff @Florian.Egloff mentioned in issue #37 (closed) 5 months ago

Florian Egloff @Florian.Egloff · 5 months ago Developer

@christian.folini I would welcome, in the big picture, to raise the fundamental question of whether people feel that the U.S. National Academy of Sciences report from 2018 is wrong in stating: “we do not, at present, have the technology to offer a secure method to support Internet voting” (NAS 2018,https://www.nap.edu/read/25120/). @Oscar.Nierstrasz referred to it in his answers to 1.1. and @Srdjan.Capkun offered scepticism in advising to use components that are still research-grade problems.

I would thus ask to open up the discussion and take a step-back: Given all the issues and debates raised, overall, do people see it as realistic and responsible to adopt internet voting in Switzerland in the short (up to 3yrs), medium (3-6yrs), long-term (6+yrs)? What would change one's answer?

(or something along those lines)

Collapse replies

Srdjan Capkun @Srdjan.Capkun · 5 months ago Developer

I would say that we can deploy secure e-voting solutions as long as we are willing to go in the direction of code voting and have voters do a bit more on their side like scanning codes with their mobile phones or typing in several codes. This would eliminate a lot of attacks and would require mainly the hardening of the counting process which could be done using a combination of technical and non-technical measures.

Christian Folini @christian.folini · 5 months ago Maintainer

Thank you Florian. We're going to have a question along these lines in the big picture. Yours is a good proposal, but I need to discuss with the team.

@Srdjan.Capkun : Don't fire all your arguments just now. Let's wait for the *real* big picture discussion.

Florian Egloff @Florian.Egloff · 5 months ago Developer

@christian.folini You opened block 10: it is again a very technical block (which is fine). I just wanted to enquire, as you included the answers of the questionnaire regarding crisis management & communication, is that NOT a topic that will be discussed?

I gave a relatively lengthy answer in https://evote-dialog.okx.ch/federal-chancellery/evote-dialogue-questionnaire/-/tree/master/responses/experts/responses/individual-responses/7.1-egloff.txt and will not repeat it here. All I would add here is that crisis management needs to be trained & operational before you run an internet voting system. It feeds into the remediation discussion in #20 (closed): you need to be able to tell voters ex-ante how you will manage a crisis when you will inform people etc.

Christian Folini @christian.folini · 5 months ago Maintainer

Your input in the questionnaire is very valuable and will be considered for further work with the cantons.

FedCh has planned to elaborate a crisis management concept that will include training and communication aspects. We will include federal specialists in this work. However, it will not be finished before the end of the dialogue. It won't be possible to discuss it on the platform because of this.

Aurore Borer [redacted] @aurore.borer · 5 months ago Maintainer

@Florian.Egloff Indeed, thank you for the feedback you gave in your answers to the questionnaire. Is there anything else you would like to add on this subject?

Collapse replies

Florian Egloff @Florian.Egloff · 5 months ago Developer

See #57 (closed) for why I think it is so crucial.

Christian Folini @christian.folini mentioned in issue #61 4 months ago