PeaceVote: absalutelly verifiable, strongly anonymous and robust cryptographic voting system by pseudonym braiding

Janis Erdmanis

May 18, 2020

Abstract

Electronic voting problem had been studied extensivelly for decades, however, no technology had achievied the same of trustworthiness as an ordinary paper ballot which is the only robust software independent voting system cabable of preserving participant privacy. A software independent, voting system designs which preserve privacy are possible if shuffling of votes is decentralized, but such system would be extremelly fragile and unpractical. In this paper I propose indentity braiding as a method for eliminating fragility of mixnet cascades for electronic voting applications and discuss a specific implementation of electronic voting for the PeaceVote electronic voting system.

1 Introduction

In ordinary paper ballot everyone is able to see that officials are acting according to agreed and fair rules as are also able to become the part of them without particular expert knowledge. On the contrary electronic voting schemes suffer from openness and transparency leaving doubt on whether the ellection officials are actually running the software that they claim to do and are not producing fake proofs to support their claims. That made Ron Rivest to claim the term software independance - a software is sad to be software independent if election outcome is independent of undetected modifications to the software which runs the elections [Rivest].

It is rahter easy to design electronic voting system which is software independent but does not preserve privacy. In a simple design each citizen would own state certified public/private key pair which forms a digital identity of the person. To run the ellections the state would publish possible option messages and would collect option certificates (signed documents) from the voters and publish them to for everyone accessable buletin board which everyone could use for verifying and couting the votes.

Preserving voters anonimity in a software independent system is hard and all known systems make a compromises. There are two issues with combining

software independance and privacy. To achieve software independance the data making a conclusive proof for accounting components of the system must be collected and published. This data shall contain votes for everyone with a cryptographical links that it was produced during the ellections by a legitimate voter. At the same time the identity of the voter casting a certain vote shall remain anonymous. We shall call this transparency-anonymity issue.

The second issue comes from execution of the protocol itself. Although the published data preserves anonimity and proves that the system had been software independent we would still face an issue of whether the software of our secure system had not been modified for recording additional information which can be used to reveal the identity of the voter. This we shall denote as security-anonymity issue, as it is possible because the voting machine (or person) has a malicious code on it.

This reasoning allows us to split software independace into two logical components. First one is security with which we shall understand machines (including humans) which execute the election software, collects votes and produces data. With transparency we shall understand all data forming a conclusive proof that the system had been software independent by producing election outcome and applies software indepnedece recursevly to the proof itself. So the trust is a combination of security, transparency and privacy which we can visualize in a Venn diagram as shown figure ....

The main difference between a paper ballot and a mainstream electronic voting solutions is by a number of machines (including humans) adversary needs to hack to compromise the proofs security-transparency (some policing officials in the paper ballot system) and to compromise the anonimity-security (an official collaborating with briber). A private company can be isolated from the society (NDA agreements, individual selection and etc.) and thus it makes it easier to hack. On the other hand hacking software all ellection officials who with their social circles cover all society is impossible without being obvious. This leads us to a idea that a secure ellection system must be open to participation so it could run synchronicaly on multiple machines at once tied to a social circles which can cover all society.

Most of the voting systems on the literature consider a rather peculiar adversary model. When the voting protocol is modeled as a process certain parts of the protocol are trusted for the purpose of verifiability. Although voter could get a assurance from the system a casting assurance (individual verifiability), and able to validate that only eligiable voters had participate, they had voted once and that the votes had been counted accuratelly (universal verifiablity) and making it end to end verifiable he/she still would need to trust the verifying system.

A similar situation is present for ensuring the anonimity of the vote, when the anonimity preserving system, for example, a mixnet cascade is often assumed to be held private to avoid malicious activities which can result in absence of election results. But if we need to trust the mixes which are private the system nevertheless of the protocol is centralised. Additional data revealing anonimity can be recorded (as long as authentification is part of a private system) and

necessary proofs of fair elections can be generated by the organization. Can we avoid thoose issues in the electronic voting system design?

The success of TOR shows that a decentralized anonymous network of mixnet cascades can exist allowing to resit govenment censourship and hiding identity as Eduward Snowden from USA. Other approaches based on mixnets, crowds, onion routing, dc nets are activelly developed and deployed [...]. When evaluating anoniity of such systems one also should take a realistic point of view of adversary who would not be able to control every ISP of the world to deanonymize the traffic. Thus with delocalization and company we can already today use quite strong anonymous sockets.

However anonymous sockets yet does not allow to avoid tracking. Unless the electronic voting is an anonymous pool where voter can vote multiple times the election voting system is going to have authetification. An interesting solution would be to use linkable ring singature scheme, sign the votes with that and deliver to the vote recorder with anonymous socket. Unfortunatelly linkable ring signatures suffer from their length and computational requirements and thus are not practical. Therefore for a practical system we shall focus on anonimization of pseudonyms which can be used for authetification of voting messages.

Anonimization of psudonyms have been studied latelly with the contex of blockchain to make transactions untracable and with context of smart vehicle networks VANET to give valuable information to authorithy while protecting privacy. To my knowledge proposed schemes do suffer from necessity to trust the mixing authorithy and sometimes also necessity to trust other partiipants in the mix. Also anonimity of thoose schemes is of a concern since network anonimization is often disregarded.

In this paper I will propose a new pseudonym anonimization scheme using only public key cryptography which are suitable as substitutes for linkable ring signatures. Then I will show how to use them for electronic voting when simultanously multiple proposals are voted on with PeaceVote system. And lastly I will touch on the issue of testing of the system for small communities, distributing software in a trusted manner while not beeing dependance on usual SSL certificates as well as how memebers can be added easally without making the sytem look complex for the voters. I would also argue that the resulting voting system is strongly sfotware independent due to the fact that anonimization of pseudonyms is done in a small batches and is strongly anonymous because of openness for particiaption.

2 Anonimization of psuedoanonymity

Imagine an ordinary elections. You go to pooling station, greet the gatekeeper who checks your pasport, go to a voting booth but then instead of putting your pick for oficials you put in the envelope your just generated pseudonym, close the envelope un deliver that to ballot box. After the ellections officials publish all pseudonyms in a buletin board (roster). The generated pseudonyms in this fashion would have two great properties namelly they would be anonymous while

on the other hand would be legitimate means to be used for signing votes. We shall call such process braiding as it does have a natural hair like structure with knots making hairs indistignuishable and also corresponds terminology used for charectarizing operations on quantum computer forming a superposition of qubits.

Braiding in essence solves the anonimous authetification issue the problem is however how can we substitute paper ballot with electronic one. It may sound like a chicken and egg problem sinece we need to solve electroninc voting to do the psuedonym braiding but it is actually not quite so. One of the great properties of signatures made by psuedonyums is that they are linkable. The linkability allows to do multiple small braids allowing to use multiple small but absoulute consensus ballots as shown in the figure .... Thoose small ballots then can be created by multiple different braiding stations which do not need to be trusted since anonimity is preserved by a lack of coordination.

To avoid a situation where participants psuedonym gets stolen in a braiding process it is necessary to make individual verifiability as part of the protocol unless other cryptographic proof is provided. That can be achieved if by the end of the ballot it is distributed for every participant to validate and sign thus forming a braid contract. That however introduces fragility in the system since any participant can refuse to follow the protocol and thus terminate the protocol while still remaining anonymous. This limits the number of participants with in the braid.

The fragility of the system can be resolved with two instruments. The first one is that the number of participants in bariding shall be kept small. The second one is using binary search technique to try isolate malicious pseudonyms from the honest ones and do the braiding with different participant sets until the braiding succeeds. Thus no participant can succeed in paralizing the system and the protocol can be cascaded to aquire large anonymity set in a robust fashion.

But who shoud do the braiding? Multiple systems could fit in as voting protocols based on dc nets and p2p networks with which one could do braiding without any authorithy. For mixnets it is necessary to trust the authorithy that no data is being recorded which can be used to reveal the anonimity. If only a single mixnet participates which is maintained by the same organization/person then the anonimity of the vote in the end is equalient to the anonimity which would be provided by a conventional mixnet electronic voting system.

The fix for that comes from the fact that we no longer need to trust the mix to do its job since in the end participants control with absolute consent whether to accept and sign or discard the braid. That allows to form a system where mixes are distributed over all internet and hosts mixing service irrelelvant of who may use it. That allows to make a braid cascade from multiple mixes and thus resolves any concerns of possible anonimity reveling data as long there are not an inicitaive to coordinate. In the end oordination of mixnet is a threat to anonimity thus any necessity of that should be eliminated.

Openness can be considered as opopisite to coordination. It is important that the voting system should be developed from the effort of the community

and as open source world had shown monolithic designs fail to attarct interest. Thus on the one hand it is important from the one hand to make the system such that it is easy to start wheeras on the other it must not loose ability to be customized without breaking off from the exisitng ecosystem. And so in the end every party would find it's place and provide mixer for other people uses.

3 PeaceVote protocol

We shall sepperate the dicussion of the proposed voting syustem in a following categories:

- setup.
- certification. How to know that you do use a correct software?
- registration. How to become a member of the organization?
- braiding
- proposing and voting
- counting and verifying
- defence.

As any voting system it needs a maintainer, administrator and protector which we shall name as guardian which can be asingle person or whoole organization. The gurdian in the PeaceVote system is responsable for a following actions:

- Start the services necessary for the vorting system.
- Making choices on cryptography used (public key system, hashing, encryption, randomness).
- If applicable, selects mixers for braiding. Sets up a number of particicpants for the braid.
- Approves new memebers for the voting system.
- Defends the system against DDOS attacks and malicious pseudonyms.

All thoose choices must be transparent and availbale for public scrutinity to change the guardian if no confidence had been expressed.

Currently, it is conisdered the best practice to issue SSL certificate on the interent service to make it available for a secure access anywhere. That however is unsuitable for a voting system where we need to consider such authorithy to be malicious or even worse selectivelly malicious. Also SSL certificates introduce a complexity for setting up the system for the guardina which I would like to make as seemless as possible.

This is where the DemeNet project comes in. In place of having a trusted authorithy it is better to trust decentralization of the communication channels.

The guardian is interested that the members of the voting system are real so that anyone should be abel to verify that the whoole community posseses human capacity which can be exercissed as power and that this power is not surpressed by unfounded participants. Currently governments are actively issuing ID cards with ability to make digital signatures. Unfortunatelly the digital signatures issued by them are only verifiable by a trust service providers which adds complexity for the guardian to verify them.

Ironically the bussiness model of social networks which uses the colllected data to individualize us is also making available data on the internet which can be used to asses the reality of the person and provides the means to establish relativelly secure communication. Currently, it thus seems sensible to use theese releatvelly secure channles to invite a person to the voting system and at the same time to deliver a trust anchor a demespec file.

The demespec file contains only essentials such as the psuedonym of the guardian, cryptographic protocols used, uuid of the self governing community (deme) which thus is a trust anchor. Since demespec file contains little information, it small and also it is not hassle of considering it to be immutable. That allows it to be easally distributed thus a potential memeber could be sure for the genuiness of the invitation from the guardian looking through his trust sources.

3.1 Setting up the voting system

To set up the voting system the guardian adds the DemeNet, PeaceVote packages and its dependancies to Julia from a trusted repository. The guardian also addscryptographic primitives for 'DemeNet' currently provided only by 'PeaceCypher' package.

The first step is to create a trust anchor for the guardian which is contained in 'demespec' file and saves that:

```
using DemeNet
using PeaceCypher

demespec = DemeSpec("PeaceDeme",:default,:PeaceCypher,:default,:PeaceCypher
save(demespec)
```

which will create a deme with name 'PeaceDeme', use 'PeaceCypher' for cryptographic primitives and say that 'PeaceVote' is a peacefounder which for us to simplify discussion currently will not be necessary. Additionally by generating a 'demespec' file a private key for the guardian is generated and placed under '.demenet/keys/uuid/maintainer'

The next step is to set up the server. First a key needs to be generated which we do as follows:

```
deme = Deme(demespec)
server = Signer(deme,'server')
```

where ‘deme‘ is a primitive which additionally with ‘demespec‘ also had loaded notary for perfoming digital signatures and cypher for establishing secure comunications with authetificated Diffie-Hellman key exchange.

The next step is to set up the configuration for the braider of the mixnet type. The mixnet braider consists of two parts - it is a gatekeeper and the mixer. The gatekeeper is responsable for authetifications with pseudonyms and is responsable to redirect connectio directly to the mixer where a new psuedonym is sent and latter redelivered back to gatekeeper sorted and forwarded to participants for forming the contract.

The guardian needs to specify the message length and the number of particpants for the braid, as well as authetification details for the mixer. Since the mixer is expected to be from a different deme we define its id with ‘DemeID‘ which contains UUID of the deme and thus cryptographic primitives can be loaded from the demespec file stored. The configuration then can be created as:

```
braiderconfig = BraiderConfig(BRAIDER_PORT,MIXER_PORT,UInt8(3),UInt8(64),SE
```

where BRAIDER_PORT and MIXER_PORT also contains information to establish a socket with server and mixer. PeaceVote only approves integers refering to local ports but with power of multiple dispatch that can be extended with external package with it’s own port type. That is particularly useful in a sense that ‘PeaceVote‘ does not need to specify whether the server is an oridnary server with ip or hidden service in the TOR. Also it is useful to specify how the ports shall be contacted anonymously, whether it is over a TOR netwok or another way which must be specified by the guardian since anonimity loves company.

The next element needed is configuration of the recorder. The recorder as name states is collecting certified applications, contracted braids, certified proposals and certified votes. The software running it is responsable to prevent redundant information to take part of the ledger. The configuration can be given as follows:

```
recorderconfig = RecorderConfig([MAINTAINER_ID,SERVER_ID],SERVER_ID,REGISTRA
```

For this service MAINTAINER_ID, SERVER_ID are trust anchors who are permited to add new memebers to the deme. The new members are coolected in REGISTRATOR_PORT who are then able to participate in the braiding. PROPOSAL_PORT currently is opppen to all members to submit their proposals (in future it will be configurble for example to a board memebers or etc.). VOTING_PORT is responsable to record all certified votes for a particular proposal. Similarly as BRAIDER_PORT the VOTING_PORT contains information on how the anonymous socket can be established.

The last part of the protocol is to decide uppon the mixer service which can be used for other demes and on the synchronization port where collected data from the recorder would be published. The full configuration for the PeaceVote thus can be formed:

```
braidchainconfig = BraidChainConfig(SERVER_ID,MIXER_PORT,SYNC_PORT,braiderc
```

Currently only a single braider is provided. It is assumed that the guardian periodically changes the braiderconfig thus preventing adversary to (even guardian) collect necessary data from mixers and guardians for revealing the identity of the psuedonym. It's possible to run multiple braiders at the same time but one then would need to deal with shceduling stategies and etc which are not important for the content of this paper of understanding the PeaceVote system.

To start the server the guardian initializes a BraidChain which contains ledger, configuration and deme. BraidChain contains all necessary elements to analyze the data collected by the ledger like to count votes and to participate in the system.

```
braidchain = BraidChain(braidchainconfig,deme)
```

It was coincios choice that the server and the user uses the same configuration data which shows that the system does not have a hidden state. A new guardian can take place of the old one with the data available in the braidchain and configure further evelution of the community at his wish. Though since demespec is immutable he would need to form a new UUID for the deme.

The last step is to start the server:

```
system = BraidChainServer(braidchain,server)
```

which would start all service necessary for the voting system. Since it does not contain any hidden data which one should keep goood care of not corrupting (except the server key) it although complex from the point of number of asynchronous services the system contains it is at the same time pure of sideeffects. That greatly reduces the maintainance costs as in case of problems one can just reinstall the server, put in back the same key (or a new one) and put in the data of the ledger which is stored by every participant everywhere preventing any corruption.

3.2 Certification

The PeaceVote protocol works because user uses the same software as the guardian. But how could users be sure that during the installation of new software a spyware/malware had not been also added?

A true answer to this question from the point of view of the guardian would be to distribute smartcards which shall be used to ensure that keys are safe and that all signatures made had been logged for accountability of additional device which is necessary to interact with the card.

Another way is to trust the software distirbutor and after the fact test whether the system is self with the trusted sources selected by the user. For example, user might receive hashes of the packages certified by guardians and the members of the demes and could compare that with hashes generated locally. That would shrink the area of attack and allow early detection of infected participants since they would certifie packages themselves and distribute the certificates which at any time could be inspected more closelly.

At the moment, however, such certification of software is not implemented since demes are assumed to be small and uncapable to perturb trustworthyness of certified authorithies which allow online and mobile banking to function in practical absence of fraud.

Another part of certification is the configuration of the PeaceVote system. If the guardian would be of curios kind he could make different settings for differrent participants which would allow to reveal identity of the pseudonyms. Thus it is important that all participants would see the same configuration and could confirm that themselves thus it must also be certified by each participant. We will return to this when we will discuss the braiding.

3.3 Registration

To take part of the deme the gurdian or it's delegated authorithy needs to sign you pseudonym. From the guardian perspective he is troubled with distinguishing genuine application over boots. In an ideal world the psudonym before sent to the guardian would be signed by a personal smartcard or by a state authorithy confirming your identity though authetification still would reduce us to the need of smartcard or other personal tooken. However ideal wolrd is not always available and if smartcards are available thoose requires developer power and some cheesy subscribtion model which toghether kills accesability.

A solution to such problem could be by using our online presence in different social networks, let it be faceboook, twitter, google, linkedin, openid and etc. which generally do provide enough information for a person to assess whether the person is real or a bot. However it is close to impossible to make a computer program which would do that automatically.

The guardian does have three options. The first one is uppon application the guardian checks the identity online and tries to establish online communication over one of the communication channels to check whether the person applying is not a bot which had borrowed a random identity and onsuch assesment would sign the certificate. The second option is that the guardian uppon a request delivers a tooken to the phone, email or etc. which can be used to register to deme. The last option is to allow everyone in and then kick of thoose which are not real.

The first option is tedius but might be necessary in case high assurance of the identity needs to ensured (investor board meeting). The last option of kicking off unreal memebers are possible by the guardian issuing order that all pseudonym braiding should restart from the their anchors. The second option of delivering a tooken uppon a request is most practical one which is what in one form or another we use on a daily basis and this is the one we shall realy on the daily basis for demenet.

The registration for the DemeNet with a tookens is currently implemented with Recruiters package. The algorithm of it is as follows:

- The guardian starts the server certifier service and the tooken reciever service.

- A potential memeber is sending application to the guardian or his delegated trusted people.
- After a review which could be also automatic if the phone number is provided or other trusted sourcce of communication the applicant receives a tooken, demespec and the information to connect to the registration server. The tooken is sent to the certifier service confidentially.
- The applicant generates a key, establishes a secure connection to the guardian certified certifieng service and sends his psuedonym and tooken.
- If tooken is found a certificate is gcreated by the server key and is added to the ledger. Additionally a log file is kept on the communication channels which were used and what information had been used as basis to issue the registartion certificate.

It is important to note that to use Recruiters the guardian is responsable to do the piping of issued certificates of the server to the braidchain. As this and other steps for configuration are so common it is recommended to use PeaceFounder package which integrates PeaceVote with Recruiters.

A great thing with tookens is that one can send invitation to people without them making to do work and if they decide to apply they can do that instantly. This is particularly great to form the demes spontaniosly just by starting a server, sending invites to the email, braid and vote. Though users would still need to install an app but if it is activelly used for multiple demes in the long term such cost would be treated as the cost of insatlling a web browser.

3.4 Braiding

At his point the memebers have been registered, they have loaded the demespec file sent to them together with invitaion and had synchronized their ledger with the guardian (one way) with the provided infomration in the invitation, for example, a port and ip address. The ledger contains BraidChainConfig certified by the guardian.

The user loads the configuration file and its first step is to gain anonimity by creating a new pseudonyms during the process of braiding. All keys for the psuedonyms are stored locally which as we will see is essential for dynamism in the voting system - braiding shall not limit voting and voting braiding. The storage unit is called KeyChain and can be initialized as follows:

```
keychain = KeyChain(deme)
```

which will load keys from .demenet/keys/uuid/* and sort them in their chonological order.

When the keys are loaded with the braidchain the user can gain anonimity by braiding which can be simply done as:

```
braid!(braidchain, keychain)
```

This simple command in the context of SynchronicBallot is performing a following algorithm:

- The user authetificates with gatekeeper in both directions with the current pseudonym and encrypts the connection.
- The gatekeeper forwards the connection to to mixer with which it has authetificated and encrypted.
- The user using the same socket is now authetificating the mixer and established encrypted connection.
- The user generates a new key and sends the pseudonym to the mixer
- The mixer collects all pseudonyms from the participants, sorts them and sends them back to the gatekeeper.
- The gatekeeper forms a braid from all collected psuedonyms. Additionally the guardian adds hash of the current ledger. (Other relevant metadata can also be added such as time up to which the braid must be published to the ledger and etc.)
- The participants validate the braid verifying whether their pseudonym is with in the braid and also and also verifying that the hash of the local ledger is the same as hash put in the braid. If validation succeeds the participant signs the braid and delivers the signature to the gatekeeper.
- If braiding went succesfull, the gatekeeper collects all signatures verifies them and publishes the braid to the braidchain. If unsuccesfull, the procedure is repeated with variations of set of participansts, perhaps of the mixer and etc.

In this proceduer the user can be the guardian certified psudonym or any anonymous psuedonym which follows this procedure. If the procedure is repeated it becomes important to anonimize the ip address of the connection which can be done with TOR or others as the guardian wishes to configure that (it can not be up to users deciding how the socket is anonimized because anonimity loves company which leads to a single option).

Additionally during the braiding participants does certify the ledger. This makes the information stored on the ledger immutable to any adversary. Additionally that ensures that all memebers are seeing the same ledger this is convinient as a place where guardian could distribute the configuration of the voting system which prohibits deanonimization attacks by a vicious guardian. Additionally the ledger is a great polace to store other information necessary for certification such as used software hashes or other demespec files and thus forming a true deme network or simply DemeNet.

3.5 Proposing and Voting

3.6 Validating and Counting

3.7 Defence

4 Discussion

- Bribing, secrecy and individual verifiability
- State insatlled malware/spyware to the devices
- The biggest mixnet due to oppenness
- Whistleblowing
- Voters need to have an application installed on their devices. Self consistency seems like good way to go. Tape like storage and external certification (one always need to be prepeared that the certifier could be the actual villan).
- Complex though the setup can be automatized making it one of the easiest system to set up. Does not require a permanent ip address. It is fine to have have a single server.
- The system in strongly software independent for voting. The ssytem does not contain hidden state and can be reset/restarted at any time with the publically available data.
- The guardians are powerless. Their wise actions and reputation is what matters for a new guardian to show up and impeach the old one.
- Additional services with DemeNet makes sense. There could be a place for a chating and messaging service. A localized search engine. A news distributor where the voting could allow for subscribers to decide uppon what content they would like to see and thus giving innitiative to pay subscriptions.
- The system scales in two directions. Though one may wish to fragment it to some regions and etc. to limit the processing power and storage needed for participants.
- In contrast to blockchain ideology, it is fine to start small. The trust is bulit from wise actions and not from competions of wasted work (PoW) and/or legitimatized inequality (PoS).
- The interent had become dystopia, full of tracking and suspicion of sincerity of information. It's time to come togehter and start to found some peace with the world around us. Its time to voice of expressing trust and breaking systematic media surpression of public opinion.

References

[1] Ben Adida. 2006. Advances in cryptographic voting systems. Ph.D. Dissertation. Massachusetts Institute of Technology, USA. Advisor(s) Ronald L. Rivest.

[2] Anderson R., Needham R. (1995) Programming Satan's computer. In: van Leeuwen J. (eds) Computer Science Today. Lecture Notes in Computer Science, vol 1000. Springer, Berlin, Heidelberg

[3] Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2), 84-90.

[4] Ruffing, T., Moreno-Sanchez, P., & Kate, A. (2014, September). Coinshuffle: Practical decentralized coin mixing for bitcoin. In European Symposium on Research in Computer Security (pp. 345-364). Springer, Cham.

[5] Serjantov, Andrei & Danezis, George. (2003). Towards an Information Theoretic Metric for Anonymity. 2482. 10.1007/3-540-36467-6_4.

[6] Danezis, G., & Diaz, C. (2008). A survey of anonymous communication channels (Vol. 27, p. 30). Technical Report MSR-TR-2008-35, Microsoft Research.

[7] Dolev, D., & Yao, A. (1983). On the security of public key protocols. IEEE Transactions on information theory, 29(2), 198-208.

[8] Christian Franck (2008). New Directions for Dining Cryptographers. MSc thesis.

[9] Golle, P., & Juels, A. (2004, May). Dining cryptographers revisited. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 456-473). Springer, Berlin, Heidelberg.

[10] Dimitris Gritzalis (2002) Secure Electronic Voting

[11] Bernhard M. et al. (2017) Public Evidence from Secret Ballots. In: Krimmer R., Volkamer M., Braun Binder N., Kersting N., Pereira O., Schürmann C. (eds) Electronic Voting. E-Vote-ID 2017. Lecture Notes in Computer Science, vol 10615. Springer, Cham

[12] Rivest, R. L. (2008). On the notion of 'software independence'in voting systems. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3759-3767.

[13] Sampigethaya, K., & Poovendran, R. (2006). A survey on mix networks and their secure applications. Proceedings of the IEEE, 94(12), 2142-2181.

[14] Dzieduszycka-Suinat, S., Ph., I.M., Kiniry, J., Zimmerman, D.M., Wagner, D., Robinson, P., & Adam (2015). The Future of Voting End-to-end Verifiable Internet Voting Specification and Feasibility Assessment Study Internet Voting Today No Guarantees End-to-end Verifiability E2e-viv.