On the Estonian Internet Voting System, IVXV, SoK and Suggestions

Abstract. The Estonian i-voting experience is probably the richest to analyze; yet there are still some society division and critics to consider. In this paper, we introduce a Systemization of Knowledge of the Estonian IVXV i-voting system and propose some security enhancements. The presented SoK discusses technical incidents, technical activities done by election observers, and different reports on the system security. We briefly trace IVXV evolution in its first years, then delves into recent fixes from 2023 till the October 2025 version connecting them to their academic sources. The paper also shades some light on two automated formal verification attempts in 2024, where researchers discovered possible privacy attacks on IVXV. We then discuss remaining vulnerabilities, focusing on the voting application problem where we alarm of a possible trojan horse voting even for verifying voters and suggest some solutions. Finally, we do seal with a whole picture recommendations and lessons learned.
Keywords: IVXV, El-Gamal Encryption, universal verifiability, trojan horses, vote buying/coercion.

1 Introduction

Estonia is a small 1.35 million population country located in east Europe who gained independence from the Soviet Union in 1991 and joined the European union in 2004 [P. Ehin et al 2019].[1] Most Estonian citizens welcomed the earlier *digital transition*; however, when it came to e-voting in 2005 there were some kind of "*notable divisions within the society between those who fully trust and those who fully distrust internet voting*" as quoted from the ***OSCE-ODIHR*** (*Organization for Security and Cooperation in Europe- Democratic Institutions and Human Rights*) **2023** report [2]. This is reflected clearly in the i-voting statistics; although the Estonian experience could be viewed as the earliest and most advanced [3], the official site [4] shows the ratio of i-votes reaching its maximum of 51% in 2023 (local Parliament), down to 41.7% in 2024 (European Parliament), then little up again to 45.8% in the just ended 2025 local elections. One can trace a long history of objection incidences mostly from right wing parties in [1] and [2/(page 8, footnotes 16&17)]; the situation was emphasized in 2023 when internet votes flipped the results for one of those parties (***EKRE***). Analysts view it as a natural echo of the society division mentioned above; i.e., it is expected for curves, [5], showing the distribution of internet votes to be

[1] The live number in 30/4/2025 is 1,347,056 (https://www.worldometers.info/world-population/estonia-population/). More detailed statistics, but dated to 8/2024, are in (https://www.stat.ee/en/find-statistics/statistics-theme/population/population-figure#);
1,127,312 "citizens" :296,268 **~ 26.28% are ethnic Russians**, (~ 1.35m-1.127m) are ≥ 1 yr residents.

completely different from paper votes curves according to each party's preference. Still, there were some complaining activities that continued persisting to the 2024 European elections [6] and the current 2025 local elections [7]; the party even called for a parliament vote to urgently suspend e-voting in 10th Nov 2025 that gained 25 approving votes and only 1 refusing vote, while the rest of 87 present members chose not to vote as shown in Fig.1. In fact, we believe the distribution of members attitude reflects even more society division that the statistics in [4]; even if the majority (61 out of 87 attending members and 101 total members) did not support "*urgent*" suspension, they cannot be classified as pro e-voting in any sense.

Fig.1. the Estonian parliament voting on a proposal to "*urgently*" suspend e-voting as taken from (https://www.facebook.com/share/p/16cvJgpr9y/, https://www.riigikogu.ee/tegevus/tooulevaade/haaletused/haaletustulemused-kohalolekukontroll/42bf2e09-0c16-4e1c-b227-478fdb5ad4b9/)

To complete the picture, it is appropriate to note that Estonia shares a border with Russia (recall from footnote 1 that ~26% citizens are ethnic Russians); according to [8] the ongoing war has put extra pressure on the country. The i-voting[2] system faced Russian attacks that authorities say were properly defended. Since most security metrics are probabilistic [9/sec. 4], the power and incentives of possible attackers have an impact on the probability of an attack to succeed; the amount of resources (money and computing power) an adversary may dedicate to an attack depends on whether it is just a party competing for a win, or a powerful country that has farsighted interests[3]. We believe the Estonian problem is further complicated by a

[2] The term *"i-voting"* refers to voting through the internet (online) which is a subset from the more general term *"e-voting"* that refers to all kinds of electronic voting that may include using Ballot Marking Devices (BMDs) at poll stations; hence, both terms can be used to describe the Estonian voting system.

[3] We say that attacking a system is computationally infeasible if the cost of computation power needed to complete the attack is more than the gain from the attack; for example, if it is an object, then it is computationally secure as long as the cost to steal it is larger than its money worth value. Mapping to elections, a rational candidate/party will not spend on hacking the elections system to forge a win more than he/she will gain from winning; however, political science researchers can measure better than the author the cost-benefit analysis for other countries (Russai as an example) towards the Estonian elections (probably differs from local elections to European Parliament elections)

dilemma of opposing factors; the parties against i-voting could be classified ideologically as nearer to Russia and the bright image of a perfect high tech election system is what Europe prefers, on the other hand overlooking the exploits raised against the system is what Russia (or any other adversary) benefits from. However, we here just notify of a possible impact and leave the analysis of measuring/estimating it to political science researchers.

After this brief preface on the Estonian election environment and the involved players, we proceed into the technical and cryptographic details; hence, the rest of the paper is organized as follows. Section 2 reports some recent important activities by the i-voting opposing community that have technical merit, while section 3 marks briefly the milestone steps through the evolution of the Estonian i-voting system[4]. Then section 4 explains in detail the current version of IVXV ending with an important attack that was fixed before the 2023 elections, while section 5 goes through recent improvements that were made in IVXV before the European elections and section 6 scans briefly the updates we recognized in the version just released on the 3rd of October. Section 7 discusses the remaining vulnerabilities of the system along with the existing research suggestions and ends with a holistic view of the system status quo that raises some general questions. Then section 8 presents a brief summary of the Estonian efforts towards the upcoming quantum computing threat. Finally, section 9 concludes the paper; Appendices contain extra details and summarizing tables.

2 Recent Opposition Activities with Technical Merit

Discussing the Estonian environment and exploring possible players in its political game with their variant power resources, is an illustrative example of how digital democracy enforced security researchers to investigate the geo-political prospective. Vice versa, this section illustrates how electronic voting partially turned the opposition-system classical debates about election results[5] into technical ones. We find observers who read papers and acquire programming skills (part of their job is to trace the election code which must be open sourced early enough for them to do so), while political parties are suspicious of what they do not know and must learn; finally, courts are cautious and sometimes confused by experts. As a general attitude, we preferred to cite social media or blog posts, not just the official statements, to give the interested reader the chance to explore the variety of comments by the Estonian citizens. Section 2.1 presents activities and complaints that accompanied the 2023 local elections, while section 2.2 covers 2024 European elections, then section 2.3

[4] Since there were many previous studies discussing earlier years of the Estonian i-voting, like [1] till 2019, this paper is more concerned about the vulnerabilities and improvements of the recent years. Our title says IVXV; a design that took place in 2017.

[5] Everywhere all the time, losers always complain and winners always defend; one of the hardest jobs is to extract the facts and the objective arguments from the equally motivated sides of the story.

goes through the currently ongoing activities of the 2025 elections that just finished while writing this version of the paper and section 2.4 is closes with concluding remarks.

2.1 The 2023 Local Elections

A technical incident that gained some publicity in 2023 elections [10] was done by the same computer scientist observer[6] in [5]; Mart Poder ***voted using his own Python code*** [11,12]. This highlighted the known fact that the voting application, which voters should download to deliver their vote, is not authenticated by the system; the OSCE/ODIHR report [2/page 8] believes the incident "*could present a cyber security risk*".

The report also mentioned *some wrong district votes*; the incident was magnified by the opposition [13,14], while described by IVXV representatives [9/ sec. 5] as a population registry problem that would have taken much longer to detect and resolve without i-voting.

Even if some exploits and/or problems were magnified, it is the authors' impression that most complaints get rejected without objective investigation based on a submission deadline (*3-days from election*); [15] is a very recent (20/6/2025) example. The OSCE/ODIHR 2025 report commented on this issue by saying [16/sec. 3.3/68 & Recom. J] "*the legislator should also address the specific complexities of internet voting when determining appropriate deadlines"*. To be fair we noticed that in many cases, as will be illustrated throughout the paper, the vulnerability gets handled and fixed silently before the following election.

Should also be mentioned in this context that Mart Poder published a poster, [17], at a scientific conference E-Vote-ID (in Oct 2023, while the elections was in March 2023); the poster questioned multiple voting as a vote coercion/buying mitigation mechanism by demonstrating it is possible to get a hard proof of the last vote through some existing system services. This will be discussed in detail with some follow-up research in sections 5.3 & 7.2.

2.2 The 2024 European Parliament Elections

We start with another rejected complaint about ***the decryption of invalid votes*** (after the elections), where the rejection was done objectively this time [18]. Among the three listed reasons, being an *observer* not an *auditor* seems to be the dominant one, where auditing is organized by the State Electoral Office in all elections. According to [19/Conclusion], generating proofs of correct decryption of invalid votes was remediated in code before June 2024, but *the file containing the decryption of invalid*

[6] The word "***observer***" is used by IVXV to acquire certain access rights during election (as opposed to "auditors"). The term "***Computer Scientist***" (taken from [10]) is rejected by IVXV representatives who describe Märt Põder as "***A hobby hacker activist***", while the OSCE report referred to who codes another voting application as "***someone with sufficient programming skills***".

votes is only accessible to auditors [19/page 22]; the reasons will be further discussed in section 5.1. The version released in October 2025 for the local elections prevents invalid votes from entering the system; however, this does not apply to other unofficial voting applications voters may use as will be discussed further in section 6.

In addition to the complaint in [15,18], and the rejecting voices [20], the European parliament elections were accompanied by some activities from the i-voting opposing community. The same observer mentioned above has developed some kind of ***shadow e-voting*** site called *virtual threshold survey* [21] encouraging citizens to vote again on it as a check (although there is no evidence of considerable participation ratio).

2.3 The 2025 Local Elections

In the current 2025 local elections, observers have prepared a voting & verification application (reveals the vote decryption), [22], and they are broadcasting a tutorial video to show voters how to use it, although we couldn't trace an effective participation ratio. In the course of doing so, they had to update their 2023 voter application[7] code to handle the new ballot format in the new version; hence, they filed a complaint to the National Electoral Committee, on October the 3rd, for the delay in publishing the system code, only at 30th of September, which does not give them enough time to study and analyze the updates. The complaint was first rejected, while the appeal to the supreme court upheld the complaint [23,24] and declared that it is "*unlawful that the State Electoral Office (RVT) failed to disclose the source code of the electronic voting system and the verification application prior to the test vote*".

Other circulating issues in 2025 elections include not publishing the voters' list [25] which could affect the tracing of wrong district votes mentioned above or the total number of eligible voters in general; the handling of the parts of the election secret key during the setup phase [26] was the subject of another rejected complaint by the supreme court [27]; also, some incidents, [28], that happened during the re-reading and integrity checking phase. The EKRE party [29], also *The Representative Association of Election Observers*[8], submitted several complaints that were rejected by the Electoral Commission then pursued to the supreme court; [30] is an observers' preliminary report that captures many details which we will discuss as we go in this section.

2.3.1 The key parts

Since the impact of key leakage will be better understood after discussing the system architecture and its underlying cryptography, we will defer it to Appendix A; we will suffice here with the incidents and the activities in response to them.

---

[7] The terms "Voting Application" and "Voter Application" both abbreviated as VA, are used interchangeably in available literature, documentation and news coverage to refer to the application that runs on the voter's device to receive the voter's choice, turn it to an encrypted digitally signed ballot, and sends it to the system.

[8] The post (http://x.com/trtram/status/1989257971663360040), just appeared while we were finalizing this manuscript, gathers the index numbers of 11 other complaints rejected by the supreme court if an interested reader wanted to trace them all at the supreme court site.

According to EKRE submitted complaint [26] and a supporting observer [27], security stickers were removed from the chips containing the key parts[9]; the chips were kept with the key shareholders inside the meeting room with them swearing they did not misuse it. The complaint was rejected on the basis that "*it is technically impossible to read and/or copy isolated chip cards from a distance"*; the observers report on the other hand, [30/point 5], stresses that the auditor ordered the key holders to put the stickers immediately, which they did not, and the auditors were mind occupied observing other simultaneous events at those 10-15 mins. In fact, the report cites a master thesis from University of Tartu, which discusses many problems associated with the auditing process in general, states as translated in [31/chapter 9] "*it may happen that the auditor may miss something important, for example being busy checking and opening security stickers at the same time"*; naturally, the argument stands valid the other way around.

It is worth mentioning in this context that there was also an older complaint about the desktop used in keys creation not being completely isolated and having some extra software downloaded onto it. A first earlier complaint granted an observer (23/2/2023) permission to see the content of the backup copy of the boot hard disk used in key creation to have full confidence there is no malware in the computer memory during key creation [32]. The observer took the photo shown in Fig.2 when the disk inspection took place (28/11/2023), then pursued the matter further to the supreme court. The supreme court responded that this could not have affected the voting results; quoting their exact words "*voting results cannot be compromised with malware, because with the help of the reading certificate issued when determining the voting results, the compromise would be revealed immediately*".[10] On Aug 2024, a commentator from IVXV team stated (in an earlier reviewers' report of an earlier version of this paper) that they admit the risk and "*it has been taken care of*". Published in the same month, the thesis in [31/chapter 9/sugg.1] recommended auditors should audit the software used to generate the key parts and must seal the system disk and its back up copy with security stickers signed by the auditors; the Estonian election act was updated in 24 May 2024 to necessitates the presence of an auditor at the generation and destruction of secret keys [33/§ 48^7/(4)], but we have no evidence that the specific mentioned recommendations were met.

[9] The EKRE party statement used the expression "*pin codes protection was removed this year*"; however, reading the full details of the incident as summarized in [30/point 5], it could be that EKRE mixed up what happened with Estonia's future plans of deploying eID chips that do not use pin codes. According to public statements, that transition (see footnotes 15,16 in section 3.1) took place on 16th Nov, i.e. after election.

[10] For the verdict statement and official progress of events, see (https://www.riigiteataja.ee/akt/328022023004, https://www.riigikohus.ee/et/lahendid/?asjaNr=5-23-40/2).

Fig.2. An image taken from [32]. According to the observer, this is the computer used in key creation which was supposed to have only an authentic Windows10 operating system, but he detected other software installed on it; DigiDoc4, Notepad++ and RamDisk.

2.3.2 The test vote reading

The election act [33/] states that a public test voting is organized before the election starts; during which, the State Electoral Office tests the casting of electronic votes, the use of the access tools set up and the determination of the results of the electronic voting and checks the integrity of the electronic voting system.[11] During the reading and integrity checking of the 2025 election, [34/mins 00:45, 15:50] and Fig.3, two issues were reported by observers [29, 35]. The first is the system failing to read the checksum of a vote, and the second is an integrity alarm declaring a duplicate vote; the voter's information was disclosed due to the errors. This was included in the rejected EKRE complaint mentioned above, [27], as one of the reasons to doubt the whole process. The election authority on the other hand released a statement, [36], described the circulating video as "*misleading*" and clarifying that the system reading error was due to a vote coming from a different voting application which did not accurately comply with the new ballot format, while the alarm came from a voter's error in repeating the same vote twice which does not reflect a problem in the system. The EKRE complaint about the whole process was dismissed by the Electoral Commission [37].

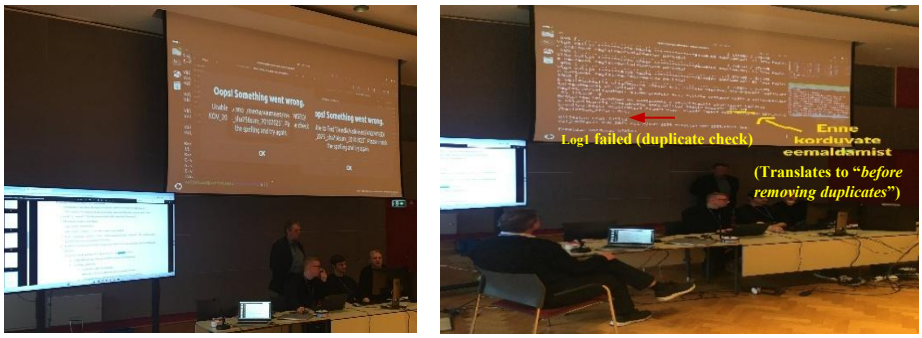
Analysis & recommendations:

We, the authors of this paper, are convinced by the official justification of the first issue; the complaint in [24,25] partially reflects the tedious format update issue discussed by the same person in [22,23]. However, the EKRE posts, [27], comments on the fact that the voter (his identity was revealed) unfairly lost his vote; the system did not send him any feedback error message[12] to revote. Hence, **we recommend applying the invalidity checks to all votes even those coming through private voter applications.**

[11] It states also that "*After the test voting, the electronic voting system is brought to the condition which existed prior to the test voting*" and this is where the key parts problem came from; the security stickers had to be removed to use the key then sealed back again.

[12] Apparently, their voting application delivered the ballot directly to the vote collector application, while the official one has a server-side part (https://github.com/valimised/ivxv/blob/published/voting/service/voting/main.go) that performs some checks and sends error messages.

We disagree though with the argument explaining the duplicate vote alarm; it should not be a problem if the same voter revoted with the same exact choice again, the added randomness (see section 4) makes the two votes encrypt to completely different cipher texts. We believe if the system here detected two duplicate ciphers, this *could possibly reflect a malicious voting application that submits the same ballot no matter what the voter chooses.* **We recommend adding a general check of duplicate ciphers exactly after removing voters' signatures and while they are still available to trace**; this will be further discussed in section. Also, since the voter's ID was already revealed during the process, we suggest checking his side of the story on the exact details of what he did; in general, **we recommend tracing individual errors even if they were not to change the election results** as part of the system debugging phase to discover any unnoticed vulnerabilities.



(3-a) min 00:45 not able to read checksum **(3-b)** min 15:51 log1 check failed due to duplicate

Fig.3. Snapshots taken from [34] before blurred as shown in [29] (the red arrow and the English caption are added by the authors of this paper)

2.3.3 The published code vs the used one

Commenting on the doubts raised by EKRE in [29,37] about "*how can be sure the published code is the one they actually run?*"; this can be achieved through file digests(fingerprints)[13]. The cryptographic hash of the published code that is claimed to run can be calculated ahead of the meeting, then calculated again during the meeting before they start the run. Again, this was part of the first suggestion in [31/chapter 9] explaining that "*it is not guaranteed that authentic software is installed on the hard drive*" and recommending an amendment should be made to make software and application installation an auditable procedure; the thesis was mainly talking in this part about the generation of the election key, but the argument applies to all used software even to the voting application code if it became open-source.

[13] Another approach of using *Repeatable/Reproducible software builds* (https://reproducible-builds.org, https://en.wikipedia.org/wiki/Reproducible_builds) has been mentioned in many places of the 2025 Estonian Cyber security reports to enable auditors from checking the authenticity of the election software and specifically the voting application; its complexity is the problem.

2.3.4 Absent e-voting

Election participation ratio between the Estonian voters (whatever through the internet or in poll stations) usually ranges between 50-60% except for the European elections 2024 was down to 37.6% participation ratio. When gathering information about the 2025 elections, we found that some voters, [38], say that when voting at poll stations, they were told that this is not their first vote; the system records an earlier electronic vote for them.

In Estonia, a poll station vote cancels the electronic vote, so in this specific case, the voter's choice was recorded correctly. However, accumulating with the previous incident, we believe this deserves an investigation; i.e., we repeat our previous recommendation of tracing the error. As a broad safeguard, **we recommend to check and alarm if the timing of poll station vote precedes the time of an e-vote for the same person**; also, **to check a random sample of e-votes using statistical techniques** (for example to contact the sampled voters and assert they actually voted in the given time).

2.4 Closure/Concluding Remarks

Finally, after navigating through the opposition' and observers' activities, and auditing tools provided by the election commission and the Estonian government in general, we present some concluding remarks.

2.4.1 Improvements

First, we acknowledge the improvements in the governmental attitude especially in this last election in Oct 2025.

- Updating the election act, since the European elections in 2024, to ensure the presence of auditors in most steps [33/§ 48].
- Publishing more parts of the code, the server-side part of the voting application.
- Although there was a complaint about the delay, the code was published earlier this time (2 weeks before the election) than during the European election last year (nearly 4 days)
- We see a lot of complaints investigated objectively compared to previous years; i.e., following the OSCE/ODIHR 2025 regulation recommendation, [16/J], of loosening the 3-days deadline.

2.4.2 Regulative Recommendation

In the regulation perspective, transparency and observation necessitates publishing the source code of the client-side voting application (VA); otherwise, there is a possibility that it could be malicious. In fact, tracing the law regulating e-voting in Estonia shows a contradiction regarding the voter application; in [33/§48^3/(7)] it states: "*Prior to the start of electronic voting, **the State Electoral Office publishes the voter application**, the vote verification application and the data necessary for ensuring the authenticity and integrity of the website on the election website*"; while in [33/§48^8/(7)] it states "***The source code for the voter application is not published***." The election authority

kind of twisted around this contradiction by publishing the server-side part and hiding the client-side part; what we, the authors of this paper, believe was not the optimal decision in this case.

2.4.3 Technical Recommendations

Trying to stand in a middle spot between [9] and the long list of complaints, we believe e-voting provides more tools to detect errors or frauds, they are just not properly used.

- A general recommendation for all kinds of errors is to trace individual errors, even if they are not to alter the election result, to find out the vulnerability behind them and whether it was invoked in other undetected cases.
- Also, a general idea is to apply statistical auditing techniques, *Risk Limiting Audits (RLAs)*, to check a random sample of votes whenever there is uncertainty about a certain risk.
- From the vote in Fig.(3-a), voters should be alarmed when submitting invalid votes; although this was part of a test procedure, the Estonian language statistical site shows 3 invalid votes in the election final results. Technically, completely preventing invalid votes could be achieved if the checks to reject invalid votes (whether in content or format) were applied to every vote entering the system even through a private voting application; this will be discussed further in section 6.2.
- Take a random sample
- Check and alarm the existence of duplicate vote ciphers, even if from different voters, as a safeguard from replay or copying attacks. This will be further illustrated in the following sections, but we mention it here since it came out from a real incident in the replay.
- Check and alarm if the timing of poll station vote precedes the time of the last e-vote. In this case, and the case above, the election authority should retrieve all available details of the problematic votes to know how this exactly happened and whether there is a non-negligible probability it was repeated; extreme possibilities like those in [39] will be further discussed in section 7 along with some suggested solution.
- A general note is the somewhat careless handling of election sensitive/crucial components or events; handling the key parts in 2023 & 2025 is an example. Also, not choosing representatives with enough knowledge of IVXV to attend the re-reading; otherwise, they would have a better understanding and handling of the errors in Fig.3 explaining the reasons behind them and that it is a good quality to have a strong error flagging mechanism even if it has some false positives (although we are not sure they are just false positives yet).
- Auditing software installation: we acknowledge the improvements in law and the presence of auditors, but there seems to be some issues in reality; the report, [30], says there were only one auditor and that the key holder asked the auditor in clear words to delay putting the security stickers for one round. Also, we are not sure whether the security stickers were signed by auditors to

exclude the probability of having two sets of security stickers; whether the software authentic installation was properly audited or not since it is repeatedly doubted in EKRE complaints and opposition posts.

2.4.4 An Academic Trusted Third Party?

Finally, we end this section by pointing out to a scientific report from ***the Cyber Security Committee of the Academy of Sciences*** that was handed to the election organizers. The complete June 2024 report was not published, but in October 2024, [40], they clarified that they *have identified* ***6 medium threats***[14] *and* ***25 low risks***. We chose to mention the 2024 report here (all reports will be further discussed in section 7) although it is not an opposition activity, since it discusses the software installation audit problems mentioned above; also, to illustrate that even academics from Estonia who could be viewed as a trusted third party admit some problems, In fact, the committee cannot be accurately considered a third party since it is led by the scientific director of the e-voting company Cybernetica; However, Cybernetica has in total only 3 out of 12 members. Hence, as a general conclusion, not all opposition complaints are exaggerated, and the authors are not just taking their side or diminishing the hard work of the IVXV team through the years. To be fair, we should stress that this report was before the improvements and/or fixes made in 2025.

3 Earlier System Evolution

As mentioned earlier, digitization has been in Estonia for more than 20 years, even before 2001, and was extended to include the private sector hand in hand with the e-government; e-ID cards existed since 2002, and electronic transactions is the casual behavior of the Estonian citizen. While details on digital system architecture and components like *Xroad*, *KSI* private blockchain seem irrelevant here, we find the e-ID key generation relevant since it is used in internet voting from its beginning in 2005. Hence, we will dedicate section 3.1 to one major event that changed a core cryptographic component of the e-ID system, **RSA;** then we will follow with a brief overview of i-voting earlier evolution till it reached its main design as IVXV in 2017.

3.1 Electronic Identity Card 2018 problem

In May 2018, Estonian authorities officially declared a persisting problem that started to appear in some rare incidences of duplicate RSA keys since 2011/2012. Citizens with problematic keys were asked to re-install the Java Applet on their cards at PPA (Police and Border Guard Board = *Politsei-ja Piirivalveamet* in Estonian language)

[14] The 6 risks (not published, https://x.com/danbogdanov/status/1802998209649762582) were first classified as "higher than small" in a meeting on 3rd of June (https://www.akadeemia.ee/akadeemia/noukogud-ja-komisjonid/kuberturvalisuse-komisjon), then a system representative commented on a review of an earlier version of this paper (14/8/2024) that all the 6 threats are of risk class medium (11-13); finally, this was confirmed publicly in October 2024.

stations. When reinstallation became more frequent, researchers started to analyze the accumulating data for the root problem. Then, it was proven that the ID card manufacturing company, ***Gemalto***, generated the RSA keys outside the chip, which violates the agreement rules and gives a chance to copy and/or repeat key pairs. A lot of interesting details on how the analysis was done can be found in [41]; more faulty keys issues[15] can be found in the same researcher's, Arnis Parsovs, PhD [42], and independently in [43]. Also, other RSA vulnerabilities were discovered in [44] which justified aborting RSA as an algorithm, not just the company.

According to [1], this was a global crisis for the company which was sued in many other countries. Spain and Slovakia [45] replaced all the physical cards, while Estonia fixed them remotely; then changed the company to ***IDEMIA*** [46] and, as recommended in [41], moved to threshold cryptography and homomorphic encryption. Currently, the Estonian i-voting system IVXV, [47], also uses ***384-bit Elliptic Curve*** Cryptography ECC with El-Gamal Encryption (section 4); yet the list of authorized votes is still signed using 2048-bit RSA key. Although [48] nullifies the effect of eID problems[16] on IVXV, we preferred to bring up those old problems to stress on the importance of studying all security issues involved with the transition to the new cards company *Thales*; a transition that the Estonian government wisely chose to make it after the election ends.[17]

3.2 Estonian i-voting before IVXV

As a preface, this section gives a condensed brief on how the Estonian i-voting system has evolved from 2005 to its final form as IVXV.
The main design theme since 2005 is a *double envelope protocol* sending voter signed ballot after first encrypted by the election public key[18] to the vote collector. Then, we mark 2 milestone step transitions [sec.1 of 49,50.51]:

In 2011, a student named *Paavo Pihelgas*[19] demonstrated a proof-of concept ballot-manipulating software that relied on the absence of an acknowledgement to the voter

[15] Including codes printed too dark that they were readable using torch, without opening envelope (happened in 2002 with the old company then again in 2018: https://news.err.ee/886313/new-id-card-issue-codes-can-be-read-using-torch-without-opening-envelope), duplicate email addresses in certificates, issuing certificates with incorrectly encoded public keys, failing to revoke certificates of deceased persons.

[16] Despite that, we chose to include this subsection in the paper to enlighten the reader that digital identity problems are part of the issues involved in the more general process of i-voting and hence digital democracy. A demonstration of this fact is that when *IDEMIA* started a transition into a new chip platform, Cosmo X, on July 2025 (http://www.id.ee/en/article/important-changes-with-the-current-estonian-card-issuer-idemia/) this necessitated some updates in IVXV 2025 version as will be discussed in section 6.

[17] *IDEMIA* contract with Estonia will end on 16th Nov 2025; *Thales* is scheduled to take over as the manufacturer of Estonian ID cards (https://www.id.ee/en/article/thales-id-card/).

[18] That's why it is called "*double envelope*"; the vote is embraced with 2 cryptographic keys: the voter key and the election key.

that his/her vote was received. Hence, [52], *the ability for voters to verify their votes* was first introduced in 2013. However, several flaws were discovered in 2014-2016, [53], that could maliciously alter the vote or the QR code.

Then, in 2017, *Cybernetica* partnered with *Smartmatic* to produce a new design, IVXV [54], with the QR verification code in its current form. A lot of improvements on IVXV since 2017 include a *vote-registration* service to guarantee no vote dropping, a *shuffling re-encryption mix-net* that cryptographically shuffle anonymized vote ballots to guarantee vote privacy, and a *Schnor based non-interactive zero-knowledge proofs (NIZKPs)* to prove the shuffled mix-net output decrypts to the same value as its original input; all of this will be detailed in the next section.

4 IVXV

In this section, we explain the design and structure of the Estonian internet voting system, IVXV, as described in the official documents [55]. Then we detail an important cryptographic attack along with its fix before the 2023 elections.

4.1 Brief Factsheet

The developing companies are ***Cybernetica-Smartmatic*** [3,55]; the voting device had to be a desktop PC, smartphone voting is about to become possible [56]; voting can be done using *mobile-ID*, *Smart-ID*, or any digital identity integrated in the *web-eID*[20]; *multiple voting* is allowed to avoid coercion or vote buying (only last vote is counted and a poll station vote overrides all i-votes); ***El-Gamal Homomorphic*** Encryption scheme is used to encrypt votes then the encrypted vote is digitally signed by the voter (double envelope); optional vote verification can be done by voters (through *QR codes* using a second mobile device) within 30 mins of voting with a max of 3 times; a ***single re-Encryption Mixnet server*** is used to scramble votes before decryption to preserve ballot secrecy. The setup phase, before the election starts, constructs the *election secret key* from 9 parts issued by the members of *the*

[19] The student filed a complaint, [50], to the Estonian Supreme court requesting to nullify internet votes in 2011 elections. The complaint was dismissed for passing the 3 days limit (https://www.riigikohus.ee/en/constitutional-judgment-3-4-1-4-11); an older example that accumulated to our impression on frequently rejecting complains based on deadline. We stress that the authors of this paper have never visited Estonia and do not have any connections with Estonian people except those we contacted as part of the Systemization of knowledge process.

[20] The IVXV version for EP-2024 included extra ***web-eID assistance service***, ***Smart-ID assistance service***,…to scale horizontally enabling the usage of different digital identities (section 2 of the architecture, sections 8.5-8.6 of the protocols in [59]). The web-eID solution (https://www.id.ee/en/article/web-eid/, https://github.com/web-eid/web-eid-system-architecture-doc) is part of the European Union web-eID project for all public key cryptography digital identities across Europe.

Election Commission of the Republic, such that decryption requires *5 out of 9 parts*[21]. Finally, after the election, there is ***an auditor application*** (could be run by anyone) that verifies the cryptographic NIZKP proofs provided by IVXV on the election published output data.

4.2 System Architecture & Voting Steps

The system architecture and voting steps are depicted in Fig.4, which could be summarized as follows

1. The voter installs the voting application, sometimes abbreviated as **VA**, on his/her PC.
2. After submitting the digital identity ID, the voting application sends to *the vote collector*, **VC**, which in turn sends to *the registration service*, **RS**, to check the eligibility of the voter to vote; if eligible replies with the candidate choices for that voter (according to district) to be displayed to the voter.
3. The voting application encrypts the voter choice using the election public key (El-Gamal encryption), adds the user signature on the encrypted vote (with the voting application running on the voter's PC and after the voter's approval, *the voting application has the right to sign a message with the voter signature*), adds also the signed *timestamp certificate*[22] received from the registration application through the vote collector *after verifying the signatures of both*, and then sends the double envelope ballot to the vote collector.
4. The vote collector application validates the voter's signature; after validation, the signature is removed, and the encrypted vote is added to the list of votes stored in the *Ballot Processor*. After voting is closed, and before sending ballots to the mix-nets, the ballot processor performs some integrity checks, removes multiple votes and votes overridden by poll station voting. Finally, the remaining list of "to be counted votes" goes through shuffling mix-nets (one *Verificatum* server)[23] to hide their original order (in local elections each district votes must be mixed separately as they will be counted separately), then verifiably decrypted at the counting phase.

---

[21] Hence comes the term "***threshold cryptography***" in the end of section 3.1, because a threshold is agreed upon from the beginning (here 5 out of 9). Relating to the key parts concerns mentioned in section 2, cryptographically, as long as malicious committee members (or leaked key parts) are ≤ 4, they cannot collude to leak the key. A reader interested in election key management of different e-voting systems can check this recent, EVoteID25, SoK (https://link.springer.com/chapter/10.1007/978-3-032-05036-6_2)

[22] When [1] was written, the timestamp certificate was used only to distinguish the last vote and to check the 30 mins verify duration. In 2024 version, we will see why in section 5.3, *the certificate* sent by RS to VC *is* a ***signed CONFIRMATION*** that contains the original request (***ORDER***) sent and signed by the VC, along with the *timestamp*.

[23] IVXV uses *Douglas Wikström's Verificatum* (https://www.verificatum.org/); the package provides a verification application, and IVXV too (and several other projects [49]).

5. The vote collector sends a verifying ***QR code***[24] to the voter for optional vote verifying (through verification application) using a second smart device.

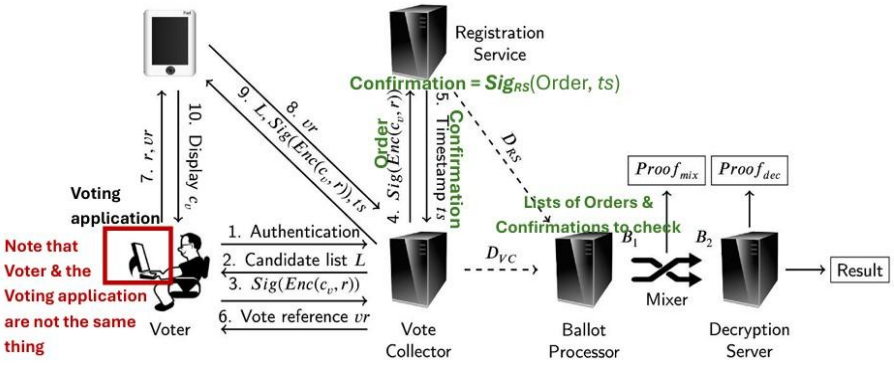


Fig.4. A diagram describing the architecture & the steps of the Estonian voting system; adopted from [1] with adding some remarks in red, and new updates in IVXV 2024 in green

4.3 Cryptographic Details of Last Fixed Attack

The exploit introduced in 2022 by [50], and fixed in 2023 by IVXV, give us a closer look into the cryptographic details of El-Gamal encryption algorithm; especially since we will mostly treat it as a black box in the rest of the paper.

-Let the election public key be "**y**" with corresponding secret key "$\mathbf{S_k}$", and "**g**" be the generator for El-Gamal encryption over the finite field **G**; hence, the equation « $\mathbf{y = g^{Sk}}$ »[25] holds.
-To encrypt a vote "**v**" the voting application generates a random number "**r**", so that the encrypted vote is $\mathbf{(C_1, C_2) = (g^r, y^r v)}$[26]
-The verification application, working instantly within 30 mins, receives "**r**" from the voting application (hidden in the QR code) and calculates $\mathbf{v = C_2 / y^r}$ where the voter is assured when the displayed "**v**" is the same "**v**" he/she voted for.
-When counting votes, the election authority uses the election secret key ($\mathbf{S_k}$) and El-Gamal encryption known equation « $\mathbf{y= g^{Sk}}$ » to calculate $\mathbf{v= C_2/((C_1)^{Sk})}$
-In the older design, the verification application only receives $\mathbf{C_2}$ from the vote collector. This gives a malicious voting application the chance to manipulate the encrypted cipher text by sending different values of $\mathbf{C_1}$ for the same $\mathbf{C_2}$. Without

[24] According to [11], there was a revealing incident of *the president vote* through his QR code: he showed it in front of cameras to encourage citizens, and someone took a snapshot. As mentioned in [9], this is not considered a privacy exploit since the President voluntarily revealed his QR-code.

[25] All the presented equations are modular arithmetic over the finite field G, and in the IVXV used elliptic curve **SecP384bR1**

[26] Now it is clear why a voter revoting with the same choice should lead to a different ballot (and the voter will sign a different cipher leading to a different signature value also); hence, we believe the duplicate vote problem in Fig.3-b reflects more than that.

checking $\mathbf{C_1}$ value, the verification application will not detect a problem/error if the voting application sent a wrong "***r***" value to the vote collector; either to drop the vote from being counted (becomes invalid), or to craft a specific ***r'*** such that $y^r v = y^{r'} v'$ to deceive the vote collector into recording ***v'*** as the voter's intended vote.

The authors found *three possible manipulations* all with *a simple fix (**only for verifying voters**):* making the vote collector send the whole encrypted pair (C_1,C_2) to the verification application which should also *verify that* $C_1 = g^r$ as was finally done [57] on 23rd Feb 2023 just before March 2023 elections. The check methodology has changed in the 2025 version to checking that the pair (C1,C2) represents a valid point on the election elliptic curve[27]; in the designated attack example; this will echo the invalidity error to the QR code if the adversary tried to drop the vote, or inform the voter that his/her vote was recorded differently (***v'*** instead of ***v***).

In 2022, the authors found it concerning [50/sec. 3.6] that such a straightforward vulnerability wasn't noticed earlier; they also criticized IVXV in general [50/sec. 4] especially in keeping the source code of the voting application secret. It turned out that the Estonian e-voting regulation law [33/§ 48^8/(7)] clearly states that "*The source code for the voter application is not published*", despite the contradiction we pointed out to in section 2.4.2; what was promoted recently (30th Sep 2025) as publishing the voter application refers to publishing only the new server-side code [58] that receives the already encrypted and signed ballot (step 3 in Fig.4) instead of communicating directly with the vote collector. The PhD in [48] says it is generally considered a national security matter; hence, we add ***an honest official voting application as a trust assumption in IVXV***, otherwise all risks associated with using a malicious voting application remain possible for the official one as long as it is not open sourced.

5 Updates and Fixes in the 2024 version

In addition to extending IVXV to support voting with more kinds of digital identities like web-id and mobile-id, [59], the IVXV version 1.9.10-EP2024 included many other improvements. This section discusses three important fixes that were committed to the official code site on 30th May 2024 before the European Parliamentary elections on 3rd of June 2024[28] and the research efforts that led to them; it also discusses providing a fingerprint authentication check (section 5.4) for the voting application. See Table.1 in Appendix B for a summary.

[27] The files in [57] now, since 30/90/2025, contains only a RPC (Remote Procedure Call) *RPC.Verify*; the real check is now done in (https://github.com/valimised/ivxv/blob/published/common/java/src/main/java/ee/ivxv/common/crypto/CorrectnessUtil.java) and if an error is detected (not a valid point) it will be seen by all verification applications.

[28] Note that the time difference here is 3 days, while it was nearly 3 weeks in 2025; this is related to the delay complaint in [24,25].

5.1 Decryption of Invalid Votes

Invalid votes are now thrown in a separate file and ***ZKPs (Zero Knowledge proofs) are generated for correct decryption of invalid votes as well***. However, election observers are not allowed to access this file or verify those proofs and there were a lot of debating and complaints about that (section 2); the issue of observers' rights persists in OSCE/ODIHR 2025 report [16/sec. 3.3/65].

Why not reveal invalid votes?
The reason is better explained scientifically as T. Kraavi did in his master thesis, [19], than by the state election service as recorded in the supreme court decision [14].
-Reasons 1&2 in [14] talk about the technical infeasibility of decrypting invalid votes after the election and how this needs parts of the secret election key. The fact is, this was already done; meaning it is feasible, and the statement was kind of *misleading the court*. The interested reader can find the scientific details in [19] and/or trace the code in ***DecryptTool.java*** file [60].
-While the 3rd reason in [14] of "*not knowing in advance what the invalid ballot contains and it may be an attack*" seems vague and not convincing, the argument is rationalized in [19]; it is the possible reveal of some information about the voter of the invalid vote, or more severe the threat of ***encoding attacks*** described in [61/ sec. 3.3 & 4.1] where an adversary can know the votes of several voters if able to submit a carefully crafted invalid vote and also view its decryption. Hence, the rational is to shrink the circle of trust into auditors only, which is almost not needed if invalid votes were rejected earlier by the system (done in 2025 as will be discussed in section 6.). Note, however, that voters can make their own voting application that allows invalid votes, and accompany it with a verification tool that enables them to view the vote decryption [22,23]; this can defeat both cautionary steps and make the system vulnerable to encoding attacks if done by an insider. In fact, tracing the *number of* ***invalid votes*** in the official statistical site [4] to be ***exactly 1*** in three local elections since 2021, increased to 2 in 2024, then to 3 in 2025 with the issue being further discussed, makes it look quite suspicious; the doubt includes anyone who can see the votes.

5.2 Integrity Checks

Another problem that was mentioned in *OSCE* 2023 report [2] is that the authorities are assumed trusted regarding *the deletion of multiple votes or ill-formed vote ballots*. Quoting their own words "*The critical step of removing the votes overwritten by another vote cast on the internet or in a polling station was not audited*", "*An insider with sufficient resources to alter the system, if able to do so undetected, could manage to control which votes are removed and therefore partially impact the results*". This was viewed by [62] as trusting the vote collector (VC) and registration applications (RS) to not collude, otherwise it would be possible to drop ballots; the *Ballot Processor* in Fig.4 could also manipulate the ballots (assumed trusted by the system). To elaborate more, yes there are decryption proofs that what goes into the *mix-nets* is

exactly what gets out of it to be finally decrypted, and yes there is the possibility to design a public independent decryption proof verifier [49], but there was no cryptographic proof for the transition from the total list of votes to the "to be counted" list of votes; what is called the *processing stage* and we believe is part of *universal verifiability*.

IVXV Adopted Solution

What we believe is partial protection from this vulnerability (see section 7.1), was integrated into the audit application of IVXV through the files, [63], ***Integritytool.java & Intcheck.py*** in May 2024; the details of the solution were published academically in [64] as of 23/12/2024. The authors state that they have contacted IVXV team with their proposed checks to detect insider risks at the processing stage, and that it was successfully deployed.

The proposed checks are applied to the Ballot Processor data which is an offline computer [64/sec. 3.A] that performs some processing on this data (step 4 in section 4.2) to then input the list of anonymized votes to the mix-nets as shown in Fig.5.

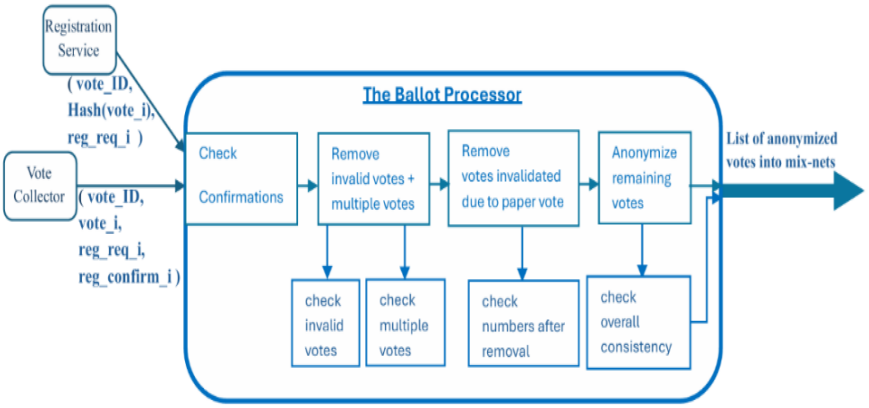


Fig.5. The (modified) processing stage after adding the *integritytool.java* file

Since the votes file (along with the necessary checksums) is cryptographically signed after each step, most examinations depend on comparing SHA256 hashes of subtotals, totals, and individual votes before and after each step. Also, *count-based validation* was needed to detect certain types of manipulations like adding the removed older multiple votes to the list of anonymized votes; i.e.,

Count (original votes file) - Count (anonymous valid votes) =
Count (multiple votes) + Count (replaced by paper) + Count (invalid votes) …Eq(1)

Note that the Ballot Processor receives the list of all votes along with any necessary confirmation checks [59/protocols.pdf/Fig.6.3]; i.e., the integrity of those cryptographic checks (like the *Count (original votes file)*) depends on trusting the vote collector and the registration application to not collude [62]. Hence, what was ***mitigated*** by [63,64] is ***the risk of a colluding Ballot Processor***; recall that it also helped in discovering other technical errors like those in Fig.3 of section 2.3.

5.3 Session ID & Time Checks

A possible attack by a malicious voting application that could deceive even verifying voters was discovered by Olivier Pereira in [51]; a malicious voting application could fake a system crash to deceive the voter to vote again. By doing so, the application can take the voter signature twice (generate another "***r***" value to construct and sign a new encrypted vote in the background); hence while showing the voter the QR code of his/her choice, the system will consider it an old vote and will use the new vote.

Previous Suggestions

- Olivier Pereira suggested a few mitigations in his paper, K. Krist's PhD [48/sec. 5.3] suggested others; none was actually adopted, and we will mention them as we go.

1. One may also recommend advising voters to double check the number of voting transactions with other available e-government services available in Estonia like *myID* service [65], especially if their device has suffered a system crash while voting.
2. Another simple safeguard from this specific problem, [66], would be *to force a time interval between votes;* the verification interval*, 30 mins,* seems a suitable choice. However, this must be accompanied by heavily warning voters to close the application then reopen again; if the voter eID remained available on the voter's PC more than 30 mins, a *Ghost Click attack* becomes possible [51] and a malicious voting application would have enough time to submit without voter's knowledge.
3. Both Pereira [51] and [67] suggested allowing to verify the last vote only, we thought of adding a flag to the QR code on whether this is the last vote or not.
4. While adding a timestamp to the QR code won't help the voter[29], a voting attempt tracking number as suggested by [51,67] will change in the revoting screen after the faked crash. Specifically, [67], the voting application should generate a session tracking number that is shown on the voting screen, added to the ballot, and checked to be new each time; the number should be embedded in the QR-code for the voter to check that it equals the corresponding number on the voting screen. However, some may view asking the voter to trace numbers as decreased usability.

-Jan Willemson argues in [9/sec. 4] that all such solutions could be used by a coercer/vote buyer to find out if the voter voted again, we believe a coerced voter is only at this risk for 30 mins (the verify duration); besides such risk exits anyway through services like *myID*, [17,68], as will be discussed shortly.

Formal Verification & Extended Pereira Attack

Published on June 2024, [67] introduced for the first time an automated formal verification of IVXV security (treating cryptographic functions as black boxes). Their

[29] Even if the voter noticed the small-time difference, could think "maybe the system records the time when I started, not when I finished"; a vote counter can't reveal following votes either.

tool rediscovered the Pereira attack and another *timing attack* that we believe is a generalization of *Ghost click* attack mentioned in [48/ref.166, 50]; the adversary stores the fake vote to submit it as late as possible to guarantee its count. We believe IVXV deployed the time difference & session ID checks suggested in [67].

IVXV Adopted Solution

A solution to the generalized attack in [67] was added on 30th May 2024. The code was updated to check the session-ID is still the same before generating the verifying QR; the code documentation [69/lines 22&105&167][30] highlights that this "***prevents reusing session ID until it is deleted from a database or expired***".

Tracing a little deeper, [59/protocols PDF/sec. 8], the ***PKIX*** *(Public Key Infrastructure X.509)*[31] timestamp protocol is used by the registration service to record the time of casting the vote, while the ***rsyslog*** service records the logging time in milliseconds which make it possible to use the *Guardtime* module to ensure the integrity of the logs.

5.4 Authenticating the Voter Application and the Pereira Attack

As [67] warns, the solution adopted in section 5.3 cannot handle the original attack in [51], since it depends on the fast sequencing of the fake crash-restart scenario; this allows a malicious voting application to use the same session ID twice and have a reasonable time difference.

According to our study of available resources, IVXV published in 2024 the fingerprint of the official voter application with a documentary, [70], guiding voters on how to check it after the download. Other than that, they sufficed with the session ID check up till now in 2025 version; they adopted the argument in [9/ sec 4] that diminishes the effect of the attack to $(1-p)^n$ for n voters where p is the probability of a voter reporting a crash when it happens.

However, we believe that the effectiveness of reporting the crash, and authenticating the official voter application to begin with, holds as an argument only **if it is assumed trusted**. Removing the trust assumption necessitates publishing the source code of the voter application (VA) for voters (and possibly observers) to review the code and then compares its digest to that of the downloaded one; otherwise, if the probability of a malicious official VA is not negligible, reporting a crash would be like reporting a crime to one of the suspects.

---

[30] The file has slightly changed in the new version released 30/9/2025; checking the session-ID and the time limit for a voting session is in the server-side of the voting application, [58], and almost every code that deals with the vote.

[31] ***PKIX*** is a timestamping protocol that enables a trusted third party (called *TMS* in [66]) to confirm the existence of a specific data at a specific point in time with its signature; can check 4 times *thisUpdate*, *nextUpdate*, *producedAt*, *revocationTime* (https://datatracker.ietf.org/doc/html/rfc6960, https//link.springer.com/referenceworkentry/10.1007/0-387-23483-7_302)

6 Improvements in the 2025 election

In addition to publishing the server-side part of the voting application, [58], and the earlier publishing of the election source code in general, IVXV 2025 version 1.10, [71], included many other updates that were committed to the official code site around 30th Sep 2025 before the local elections on 19th of Oct 2025. This section covers technical implementation updates like changing the ballot vote format and the way El-Gamal encryption is verified, then moves to discuss in detail the significant improvements of using Range Proofs to prevent invalid votes from entering the system. These improvements are also included in Table.1 of Appendix B.

6.1 Technical Enhancements

The updates in include, as detected by the authors of this paper, but maybe not limited to:

- Changing the ballot format, [58], in a better way to include the checksums integrity checks needed in section 5.2 and dynamic qualifying services (like the voter eligibility to vote or any other future purposes); this was reflected in the extra effort [22] needed to adjust their private VA, and the first error in Fig. (3-a) also came from it.
- The dynamic qualifying services in [58] includes also a check of the voting duration time to enforce a tighter time limit on the session ID; i.e., further enhancement of the session ID tracking safeguard discussed in section 5.3.
- Some changes to handle the new eID chip platform discussed in footnote 15; one can also trace future plans (TODO parts of code) for the upcoming new *Thales* cards (footnote 16).
- Switching the regular check of El-Gamal encryption pair (the vote), which involves exponent raising, to the more efficient check of it constituting a valid point on ECC the Election Elliptic Curve *SecP384bR1*[32], [72], as discussed in section 4.3
- Enhancing the mix-net shuffling code, [73], to add support for the elliptic curve cryptography methodology introduced above. We could not accurately trace repeating the mixing into more extra rounds as previously criticized; however, the new protocol file [71/protocols.pdf/secs 7.1,10.1] explains that in this local election votes are first grouped according to district before mixing.

[32] The interested reader of exploring different approaches can refer to this recent study that investigates efficient and portable ways to produce verifiably encrypted bits as usually done in homomorphic voting (Henri Devillez, Olivier Pereira and Thomas Peters, 11 Nov 2025, "*How to Verifiably Encrypt Many Bits for an Election?*", https://link.springer.com/article/10.1007/s10207-025-01140-x)

6.2 Range Proofs

To eliminate any complaints (like [14]) and/or risks (ex. [61]) accompanying invalid votes, it is much safer to prevent them from reaching the Ballot processor at all. The thesis in [19], followed by the scientific report [74], suggested the use of a Zero Knowledge Proof (***ZKP***) check by the Vote collector application to check the validity of the vote it receives from the voting application without revealing it and hence reject invalid votes before being added to the list of votes.
-The thesis preferred the use of ***Range Proofs*** as opposed to ***Set-Membership proofs*** for their simplicity and suggested some mitigations to the discontinuity of the set of vote choices. The authors proposed Range Proofs that are based on ***Bullet Proofs & Pedersen Commitments***, since they depend on the Discrete Logarithm problem like El-Gamal encryption used in IVXV. They considered general purpose SNARKs (Succinct Non-Interactive ARgument of Knowledge) based on polynomial commitments not suitable for El-Gamal based voting systems. The interested reader can find more discussion of this ZK specific choice on the extended version of this paper [75]; the Range proofs presented above are what is implemented in the 2025 IVXV version (with TODO parts to optimize proof aggregation for larger ballots). The validity check can be clearly traced in the published server-side part of the voting application, [58], which performs different checks (including Range Proofs) on the vote before sending it to the vote collector. The Range Proof check was also added to the verification code because a private voting application can still bypass the code in [58] and send votes (possibly invalid) to the vote collector directly; this fact was proven through the official statistics, [4], showing 3 invalid votes in the 2025 elections. We believe the risk could be eliminated completely if range proofs were deployed through the vote collector to check even votes from other private voting applications.

7 Remaining Issues & Further Research suggestions

This section collects existing research that either points to remaining vulnerabilities in the IVXV Estonian i-voting system or suggests further enhancements to strengths its security, then seals with a holistic view of the status quo. Table.2 in Appendix B summarizes the findings of this section.

7.1 Adding NIZKP (online) to remove trust assumptions

This was first suggested in 2022 by [61]; to add a *NIZKP of knowledge* to each encrypted vote as a protection from privacy attacks. The 2024 version of IVXV (1.9.10) partially responded; the Registration Service now sends *hash(vote)* to the Ballot Processor (Figs. 4&5) as explained in section 5.2. Tracing the code [63/line 239], vote hashes are stored in a *Treebag* which represents a binary search tree data structure in Java; i.e., even if it reflected a Merkle tree design, the number of nodes in a Merkle tree (votes) is not cryptographically verified. However, it is not just privacy attacks, and it is not just the Ballot Processor that we need to remove trust from.

In 2022, [48/sec.5.1.2, 44] discussed IVXV trust assumptions including software components and key holders; in 2023, [62] acknowledged the risk if the Vote Collector (VC) and the Registration Service (RS) colluded together. The 2024 [67] usage of automated formal verification methods [76] asserted the same result; when analyzing IVXV End-to-End-Verifiability (EEV) and privacy under 9 different combinations of trust assumptions [67/Table1] for an adversary controlling more than "some voters' credentials", EEV^{+} (that maps to current IVXV status) fails if 2 of (VC, RS, timestamping service TMS) colluded. Also, both the OSCE/ODIHR 2023 [2] and 2025 [9/sec. 3.3/28] reports mentioned persisting political parties concerns regarding insiders' trust.

Hence, the 2025 E-Vote-ID conference contained two suggestions of using Vector Commitments (usually referred to as Verkle Trees); the first [77] is a poster that suggests the use of Verkle trees based on KZG commitments, and the second is a paper [78] suggesting those based on Pedersen commitments. Readers interested in deep cryptographic analysis of the differences can find it in [79]; however, we anticipate that the next version of IVXV will deploy Pedersen based vector commitments. The suggestion in [78] aligns with the system's previous choice in Range Proofs, the authors are from the IVXV team, and they claim faster simulation time of less than 1 sec.

7.2 Integrity-Coercion tradeoff and other existing sources of information

The PhD in [48] was the first resource we encountered, 2022, concerned with the tradeoff between integrity vs coercion resistance when discussing mitigations for handling a corrupted voter device. Also, [61] demonstrated (through introducing the possible privacy attacks discussed in section 5.1) that *IVXV is vulnerable to attacks against vote privacy in those threat scenarios that were considered for it originally*. In addition, [67/ Fig.1.D] usage of automated formal verification methods discovered a new attack against privacy using some form of replay attacks.[33]

Getting back to the Estonian research, in 2023 M. Poder presented it clearly in a poster [17] how the *myID* service [65] can provide hard copy evidence of the last e-vote in Estonia which destroys the multiple voting defense against vote coercers/buyers (recall the discussion in section 5.3) and contradicts with the election act [33/§48^{6}/(9)]. Accordingly, in an earlier version of this paper, we raised the broader question of *to what limit can the information provided by general purpose activity logs of digital identities* (in any country that uses digital identities in online voting) *help vote buyers/coercers* in catching voters who try to deceive them. Then, in Oct 2025 [68] discussed the specific Estonian case and suggested different possible solutions like making election transactions secret or obfuscated or even using a

[33] The attack copies b=(C,s) of an honest voter into b'=(C,s') where s,s' are the signatures of "C" the ballot as a whole C=$(\mathbf{g^r}, \mathbf{y^r v})$=(C1,C2) in the terminology used in this paper; they say this strategy of minor biases in the outcome could be generalized leading to quantifiable privacy violation [67/ref.44].

special ID card for e-voting that doesn't allow such integrity checks on its transactions.

A final remark on this context is that suggestions that involve applying aggregating queries as consistency checks on the election results numbers, like [77], can still be deployed with any of the suggestions in [68]. Those queries are assumed to use zero knowledge techniques; i.e., they do not necessitate revealing the raw transactions data.

7.3 Automated Formal Verification and Mix-net Servers

Although formal verification tools in general, like [76,80], treat cryptographic details as black boxes that one may diminish their value in testing IVXV, two independent attempts were introduced in June 2024 from Luxembourg and France; to our knowledge, the literature does not hold automated formal verification analysis of IVXV done by researcher from Estonia. The first, [67], was discussed in sections 5.3 and 7.2; the second [81] was mainly concerned about mix-net servers. The authors used *ProVerif* , [80], to discover [81/ sec 4.3 & Table 5] the same privacy attacks in [61] showing that they can only be defeated using a strong NIZKP for each vote (section 7.1); the paper also contains a comparison analysis between exponentiation mix-nets and re-encryption (like IVXV) mix-nets for the interested reader.

It is also worth discussing that IVXV uses only one single mix-net server[34]. Although *Verificatum* uses in its documentation, [49/ref.1], an example of 3 mix-net servers, IVXV uses its own implementation of one single server; the argument is simplicity and that distributing the key into parts provides an additional security layer. As a basis of comparison, the Swiss-Post i-voting system used in Switzerland has 4 mix-net servers and [82] discovered that NIZK proofs must be verified after each server not just at the end of the whole shuffling process.

7.4 The Voting/Voter Application (VA)

We believe the voting application is the most persisting and/or serious integrity problem in the Estonian i-voting system. All risks associated by previous literature with a corrupted voter device, or a corrupted communication channel, are perfectly feasible with a malicious voting application and even more.

7.4.1 Associated Risks

In addition to the usability issues regarding the available authentication method added in 2024, [70], of the official VA; as long as the code is not published, it is based on a trust assumption of it not being malicious. Getting back to the student complaint in footnote 18 of section 3.2, the QR code verification was an enough

[34] We thank a previous anonymous reviewer of an earlier version of this paper for pinpointing that weakness point.

solution for ***individual verifiability***[35], but with more than 90% non-verifying it is not enough evidence for ***universal verifiability***; the election result may be very well altered by a malicious voting application. In fact, anyone who has enough access rights to edit a few lines of the secret code, or upload a different file to the site, can do that. If non-verifying voters can be identified somehow, for example through social engineering, traditional techniques for Public Bulletin Board (PBB) attacks [83] can be used here.

Even for verifying voters, the current situation leaves it as an open challenge for adversaries to design a malicious VA that can invoke any hidden undiscovered exploits to deceive them; a simple example is ***a trojan horse*** *application that deceives the voter to take his/her signature for another purpose* like online buying transaction or account registration transaction. Although [9] argues to diminish the effect of the attack in [51] along with the threats in [2], what if the attack was performed using any other application the voter downloaded after casting vote; what is the probability of detecting the error in this case.

In addition, *new technologies like online coding through blockchains and smart contracts*, [84], have introduced more capabilities to render automated large-scale[36] attacks and/or execution attests for vote buyers/coercers, [39], feasible. The attack in [39] could be viewed as an extension of the *copy attack* presented in [67] (which we discussed in section 7.2 and footnote 32).

7.4.2 Suggested Solutions

-The only defenses we find possible from the trojan horse case, without solving the voting application problems, is for the Estonian government to separate i-voting transactions. This could happen **by adding a meta data field that check the application category** so that IVXV does not accept transactions from applications registered as gaming, buying,...etc; also, could be achieved through the suggestion in [68] of **using a separate ID card for e-voting transactions**.

---

[35] The term "***individual verifiability***" refers to proving the correctness of each voter's vote with overwhelming probability; this prove is usually handled by proving 3 stages 1-"*casted as intended*", 2-"*recorded as casted*", 3-"*counted as recorded*" (sometimes 2 steps are mixed like proving 1 then proving "*casted as counted*",…etc). After the fix in section 4.3 (sending C_1,C_2 not just C_2, the Estonian i-voting has no detected problem in individual verifiability. ***Universal verifiability*** on the other hand refers to the correctness of the election results, whether it could have been altered or not; for example, many researchers do not consider tracing individual errors, which we suggested in sections 2.3&2.4, part of universal verifiability if those errors were not to alter the election result (the difference between the winner and the loser is more than the number of doubted vote).

[36] Previous researchers like [48] (based on [48/ref.166]) counted on observable anomalies (like financial theft if the digital ID is compromised) to argue that those kinds of attacks cannot go unnoticed if performed on a large-scale; we believe there are possible adversaries that might be interested only in altering election results, and others who may defer any financial malicious activities till the election ends. Besides, [48/ref.166] was written in 2014 before smart contracts provided online execution [84/sec.3, sec.5].

-We suggest as safeguard to protect from the copy attack, and from situations like that in Fig. (3-b) whatever caused it, is **to check for equal values in the list of encrypted votes** and raise an alarm if found any. This could reflect an attack that repeats an adversarial encrypted choice through a set of compromised voters' credentials (whether bought, coerced, or hacked); note that this protects only from the simplest form of the attack in [39] if the adversary did not add a fresh randomness to each clone to be $\mathbf{(g^{r+rnew}, y^{r+new} v)}$.

-Other suggested solutions are mostly straight forward; change the law to **publish the official voting application** with a more **simplified batched authentication** [85] to check the published code is the used one; allow only **private VAs** that **are pre-registered** and hence **authenticated** and examined for malicious code before the election; encourage voters to verify; **alarm voters** if their vote was casted using a private VA and alarm voters of a new voting transaction in general; also, connected to the alarm, encourage voters to vote in the first place (maybe by **adding a "reject all" or "boycott" option**) and/or encourage **nonvoters to check if someone forged a vote for them** using services like myID.

Since the risks can be magnified for absent voters, as there is no QR code to check (recall [38] from section 2), we dedicated Appendix C to discuss this case. We just mention here the somewhat different approach to IVXV for observers to take ***a* random sample** from those who did not vote; the sample is then **to be checked against the election results** using traditional known statistical methods like different *Risk Limiting Audits (RLA)* techniques.

7.5 Cyber Security Committee 2025 Report

We appreciate the Cyber Security Committee step in Spring 2025, [86], to change its assessment methodology to include *risks that become more serious due to the combined effects of several hazards* which perfectly describes the attack in [39] and the trojan horse attack. Navigating the report, we find 5 out of 9 threats that were considered noteworthy threats are about the *voting application* (section 7.5.1). The other four noteworthy threats are the *quantum* threat we will discuss in section 8, the *developer dependence* threat of using a limited number of developers for the whole system code especially with the absence of bug bounties and the secrecy of some of its parts, and finally 2 risks associated with poll station voting canceling an e-vote (section 7.5.2).

7.5.1 Noteworthy threats concerning the Voter Application (VA)

- They admit that the distribution of mock voter apps undermines election integrity.
- They have also assessed the threat of both open and closed source code voter application (VA) and found that their impact is similar, but the possibility of a threat event is higher in the case of closed source code. They describe the trust problem as the threat of "*code written into the voter application that manipulates the voter's expression of intent*". The report also

added the risk of not detecting programming errors, security vulnerabilities, back doors, covert channels, and side channels. When the report considers that publishing the code will make it easier to produce a malicious VA, what sometimes referred to as a *rogue application*, we say that this threat already exists with the current existence of other voting applications.

- They examined added threats when allowing voting through smart phones; whether to download the VA from an official site or make private contracts with Google play and Apple store to prevent fake applications. As a less significant threat, the report also discussed whether smart phone voting is worth the effort or is it better to dedicate the effort to enhance the security of the current system.
- However, the report did not mention the risk we mentioned above of injecting votes through trojan horses embedded in other purpose applications that involves taking the voter signature; we believe this is a significantly feasible risk that will be further magnified if smartphone voting is applied (people use their smartphones more frequently in miscellaneous apps).

7.5.2 Threats from the connection with poll station voting

Recall the absent e-voting incident in section 2.3.4, there are possible reverse risks; the thought of "*what if the voter is lying, or the poll station officer is lying*?" rings a bell on providing guarantees that poll station officers do not alter the election results by altering the names of poll station voters. Presumably, there are enough guarantees that the number of paper votes in a voting booth equals the number of names marked as voted there; what guarantees can prevent a malicious poll station worker from marking the wrong individuals.

A possible scenario is to mark some e-voters if they are known in the district of their support to a certain candidate who encourages e-voting, in place of another candidate supporters who can now double their votes through e-voting. Some may minimize such probability as unlikely, because in the Estonian case complainers are those who advise their candidates to vote in poll stations; however, politics are sometimes deceiving.

Protection suggested in the academy report

To use a notification channel (ex.: an email) to inform the voter of a poll station vote; this will work for wrongly canceled votes. Then we can use the same probability argument in [9/sec.4]; if a voter will report a forgery to the election authority with probability "*p*", a malicious poll worker who altered *n* votes will be detected with probability (1- p^n). However, we remind of the possibility of deliberately delaying the side channel messages [51] warned from.

We suggest the following

1. Before deleting e-votes corresponding to poll station votes, check for anomalies like an e-vote with time stamp larger than the timing of the corresponding poll station vote; this involves accurate timing of poll station

votes, which may hesitate malicious workers from conducting the attack as an added advantage. Another anomaly could be that a significant ratio of cancelled votes came from a certain poll station.

2. Take a random sample from cancelled votes and check that they had really voted in the corresponding poll station; *RLA* measures may be applied on the sample result.

7.6 Quantum Computing Risk

The report in [86], and earlier Estonian efforts [87 included in its new assessment methodology *future risks* and featured the quantum threat as *noteworthy* threat. Quantum Computing (QC) devices, whenever they exist, can break the cryptography used in IVXV which necessitates the migration to new post quantum cryptography (PQC); however, the transition is expected to take years for many reasons[37]. The report also shares our warnings in earlier versions of this paper from the privacy violation if old election data (usually destructed after a month) were leaked to be decrypted when QC is available; a concept cryptographers call[38] ***Harvest Now Decrypt Later (HNDL).*** See Appendix D for more details.

7.7 AI Risks

The previous report, [86], only mentioned as a less significant the possibility of an AI installed on the voter device could infer and leak the e-vote. However, in light of the recent attacks conducted solely by AI[39], *we classify AI attacks as a* ***significant threat***; the risks include identifying exploits for attackers, deceiving a voter to sign an unintended vote, or performing a general attack on the system. We deter the detailed assessment of such risks to future work.

7.8 Final thoughts on the status quo

In 2023, [62] analyzed IVXV *public information* as *satisfying* only *1 out of 9* (minimal restriction on disclosure of vulnerabilities) *quality metrics*[40] and warned from the possible existence of hidden vulnerabilities. In June 2024 mins of meeting [87] and secret report, the commission classified the 6 security risks found as "Higher than low"; in October 2024 they published in a conference [40,88] that all ***6 risks are medium*** and announced the existence of another ***25 low risks***. Their 3-5 (out of 6) focus of development areas supports our discussion about revealing and

[37] https://e-estonia.com/cybernetica-post-quantum-cryptography-joins-to-menu

[38] https://a16zcrypto.com/posts/podcast/quantum-computing-what-when-where-how-facts-vs-fiction/

[39] https://www.cybersecuritydive.com/news/anthropic-state-actor-ai-tool-espoinage/805550/, https://www.anthropic.com/news/disrupting-AI-espionage

[40] Used quality metrics were defined in an earlier paper by the same authors (FC'21, *New standards for e-voting systems: Reflections on source code examinations*)

authenticating the voting application; this was assessed again in the new report released Sep 2025, [86], as discussed in section 7.4.3.

This proves that IVXV is continuously evolving and improving with time; we traced many improvements in IVXV 1.9.10-EP2024 and then in 1.10-E2025.[41] However, here comes another issue; if we have missed some updates, we believe it is because "***IVXV does not welcome strangers***" as a general attitude and favors researchers from Cybernetica team or the Estonian Universities as the most outer circle. The last English language PDF files are dated to the 2022 version which can be traced as the cited reference in all 2024 outsiders' research like [67,81]; we had to use browsers translators, trace GitHub commits, social media posts, use AI and contact involved personnel. This was also illustrated through the mix-nets available material; we could not find any document that states clearly the number of used mix-net servers, which turned out to be only one, especially that the cited documentation of *Verifactum* in [49] uses a 3-servers example for demonstration.

We seal this section by raising a question on ***whether e-voting systems are classified as a political national security matter countries ought to be discreet about, or a security research area and a cryptanalysis topic open for researchers?*** The authors of this paper are not against e-voting; we believe that although it introduces new risks, it still provides means to detect what cannot be detected using paper voting only. However, this necessitates open discussions as Estonian academics believe, [40,86,89], though real actions sometimes contradict and implies the national security matter convention mentioned in [48].

8 Summary & Conclusions

In this paper we gave a political and technological historical brief on the development and status quo of the Estonian internet voting system IVXV. We have also shed some light on the technical activities and complaints of the opposing community against e-voting in Estonia along with analyzing the incidents that triggered them and advising some recommendations. We explained the current system architecture and surveyed available material from academic literature and different other resources to cover reported attacks and/or vulnerabilities and how they were fixed by the system throughout the years, with extra focus on the 2024 and 2025 versions.

Then we discussed the remaining issues and the suggested research solutions; we addressed the concerns of opposing parties in their meeting with ODHIR [16/sec.3.3/28]; we alarmed of possible trojan horse attacks even for verifying voters, and warned about the feasibility of orchestrating large-scale attacks with the existence

[41] In 2024, GitHub showed 897 changed files with 34,059 additions & 10,830 deletions; in 2025, it shows 732 changed files with 11,532 added lines and 39,893 lines removed. Translating [59], dated 2024, a lot of work was done in integrating different kinds of IDs and in coordinating with XRoad service (X-tree). A whole section is dedicated to Registration Service, [59/protocols/sec.6]; the interaction between online (RIA), offline (RVT) and other IVXV services. The same applies for the 2025 version.

of new technologies like online coding and code attestation. Hence, we recommend dealing with published easily authenticated voting applications and adding strong NIZKP (not just checksums) to each vote. Finally, we did not neglect to mention the Estonian efforts towards the upcoming quantum computing threat, and to warn from new AI possible risks although we leave the exhaustive analysis to a future work.

The technical suggestions of this paper are summarized in Table 2 of Appendix B; however, general suggestions include tracing individual errors even if they are not to alter the election results and performing a reading process before closing the poll station voting to give a chance for error votes to express their opinion. Then, we strongly advise the Estonian government to publish the voting application code even if they had to change the regulation for that. Last but not least, for the electoral commission and the IVXV team to be more open to scientific discussions as stated in their own recommendations [40,86,89].

Finally, we hope also our reader has realized the so many steps and details involved in digitizing elections; from regulations to chip cards used for digital IDs to auditing from the setup phase and software installation up till the verification; from cryptography & cryptanalysis to software developments to political game theory sometimes; …etc. Yet the journey is worth the trip; for elections as it provides extra tools to detect and trace malicious behavior, and for researchers as it embraces a rich and diverse field material. While writing this paper, we navigated through, and gained knowledge from, a variety of different research areas; there is geopolitical factors and there effect on the security game, cryptography with cryptanalysis, zero knowledge proofs with their implementation details, automated formal verification methods, blockchains online smart contracts, hardware related details like execution attestation and informing users, repeatable software builds, code tracing for those many files, and finally the secrecy of the voting application retrieved an old read in Bruce Scheiener "*Applied Cryptography*" book of an old debate about the secrecy of cryptographic algorithms used by governments. We hope readers of this paper from those different research areas have found something useful they can build upon.

Acknowledgements

I'm grateful to everyone participated in making me acquire this level of knowledge & research skills: from my graduation faculty and postgraduate supervisors nearly 30 yrs ago, to everyone who has put their work free online (Berkeley ZKP MOOC, HAL, iacr, arXiv, E-Vote-ID, Tartu & Tallinn Universities); also, many useful links were reached through talking to people on social media platforms. Finally, this work benefited from AI companions (grok, Gemini, and Copilot) in 2 ways; the first is asserting thoughts as discussing with a colleague researcher, the second is repeatedly generating AI review reports to discover weakness points and prepare a revised version.

References

1. Piret Ehin, Mihkel Solvak, Jan Willemson, and Priit Vinkel, "*Internet voting in Estonia 2005–2019: Evidence from eleven elections*", Oct 2022; https://doi.org/10.1016/j.giq.2022.101718; https://www.sciencedirect.com/science/article/pii/S0740624X2200051X
2. OSCE/ODHIR 2023 report on Estonian Internet Voting, https://osce.org/files/f/documents/f/f/551179_0.pdf
3. https://www.smartmatic.com/featured-case-studies/estonia-the-worlds-longest-standing-most-advanced-internet-voting-solution/; last accessed 30/6/2024.
4. https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia; last accessed 30/4/2025.https://gafgaf.infoaed.ee/en/posts/great-divide-in-evoting/; last accessed 14/3/2024.
5. https://ausadvalimised.ee/docs/yhisavaldus2023/; and newer petitions in 2024: https://ausadvalimised.ee/ei-lepi-vaadeldamatusega/, last accessed 13/6/2024; https://x.com/ausadvalimised/status/1808854585597108552, last accessed 5/7/2024.
6. AGO Samson' article *Postimees* refused to publish, 16/10/2025, https://www.facebook.com/share/p/17bctWG8si/; last accessed 27/10/2025
7. Alexander Martin, 13/3/2023, https://therecord.media/estonia-cyberattack-parliamentary-elections, last accessed 8/7/2025.
8. Jan Willemson, "*Recommendations to OSCE/ODIHR (on how to give better recommendations for Internet voting)*", E-Vote-ID 2025, LNCS pp. 208-223, Oct 2025, https://link.springer.com/chapter/10.1007/978-3-032-05036-6_13
9. "*A computer scientist made available the code for e-elections, which the electoral service has so far been fiercely hiding*", https://digi.geenius.ee/eksklusiiv/arvutiteadlane-tegi-kattesaadavaks-e-valimiste-koodi-mida-valimisteenistus-on-seni-kiivalt-varjanud/; last accessed 2/1/2024.
10. https://gafgaf.infoaed.ee/en/posts/perils-of-electronic-voting/; last accessed 4/1/2024.
11. https://media.ccc.de/v/37c3-12298-should_e-voting_experience_of_estonia_be_copied#t=965; last accessed 15/1/2024.
12. Post Times, "*The use of e-voting should be limited*", https://arvamus.postimees.ee/7974894/mart-poder-e-haaletuse-kasutust-tuleks-piirata; last accessed 13/3/2024.
13. Election Commission of the Republic, "*Resolution of Andres Alla's complaint*", 21.06.2024 No. 14, https://www.riigiteataja.ee/akt/322062024003; last accessed 5/7/2024.
14. National Electoral Commission, "*Resolving Märt Põdra's Complaint*", 20/6/2025, https://www.riigiteataja.ee/akt/321062025005; last accessed 29/5/2025.
15. OSCE/ODHIR, "*Opinion on the Regulations of Internet Voting in Estonia*", 17 June 2025, https://osce.org/files/f/documents/e/a/593435.pdf
16. Mart Poder, "*Hard evidence of an e-vote in the IVXV protocol*", Lightning talk, E-Vote-ID 2023 (2023). https://infoaed.ee/proof2023/; last accessed 28/10/2025.
17. https://www.valimised.ee/en/internet-voting/observing-auditing-testing; last accessed 5/7/2024.
18. Taaniel Kraavi, supervised by Jan Willemson, "*Proving Vote Correctness in the Estonian Internet Voting System*", Master thesis, Tallinn University of Technology, June 2024, https://digikogu.taltech.ee/et/Download/ffdf0de1e58d455ba3d484400c9123fc.pdf
19. Ago Samoson, "*The developers of our e-election system could admit their strategic mistake in order to prevent the worst*", 17/3/2024; https://arvamus.postimees.ee/7981474/ago-samoson-valimishavingut-tuleb-ennetada, last

accessed 9/7/2024; on 17/3/2025, https://arvamus.postimees.ee/8211830/ago-samoson-valimised-ehk-e-mang-koduvaljakul, last accessed

20. "*A transparent digital ballot box can be tried in the e-voting threshold survey*", https://ausadvalimised.ee/uuenduslik-exitpoll/; https://github.com/infoaed/pseudovote-euro24/tree/JUNE5TH2024; last accessed 5/7/2024.
21. "Independent Vote Verification tool for IVXV protocol of Estonian e-voting 2023 and beyond", https://infoaed.ee/vote2025/; GitHub: https://github.com/infoaed/kryptogramm; video: https://drive.google.com/drive/mobile/folders/1grTuBWJDgYKVqWbVxIy6VdlyhJiJ95Co?usp=sharing; last accessed 21/10/2025.
22. The original Supreme Court Verdict about the delay in publishing the opensource parts of the voting system; Estonian language: https://www.riigikohus.ee/et/lahendid?asjaNr=5-25-55/4; Copilot English Translation: https://anonymous.4open.science/r/SoK_Estonia_IVXV_EVoteID-C2E4/court_verdict_on_delay.pdf
23. The statement provided by the court on the same delay verdict, https://www.riigikohus.ee/et/uudiste-arhiiv/riigikohus-e-haaletamise-lahtekood-tuleb-avaldada-enne-proovihaaletust; last accessed 31/10/2025
24. AGO Samson about not publishing the voters' list, 9/10/2025, https://www.facebook.com/share/p/1Bks6RqMmV/; last accessed 23/11/2025.
25. EKRE rejected complaint to the State Court, 24/10/2025, https://www.facebook.com/share/p/14N3t8JjJfS/; last accessed 27/10/2025
26. Supreme court decision about the key leakage issue, https://www.riigikohus.ee/et/lahendid/?asjaNr=5-25-57/7; last
27. M. Poder, "*Why is the electoral service afraid to admit mistakes*", 22/10/2025 https://gafgaf.infoaed.ee/posts/oops-somethin-went-wrong/; last accessed 27/10/2025
28. Martin Helme, head of EKRE, on 7/11/2025, https://uueduudised.ee/martin-helme-need-ei-ole-tosiseltvoetavad-valimised-mis-meil-toimuvad/; last accessed 7/11/2025
29. Preliminary Report of Observers, "*Secrecy of vote in local government elections not guaranteed*", https://ausadvalimised.ee/docs/kov2025-eelraport/; last accessed 23/11/2025.
30. Lauri Ütsik, supervised by Kristjan Krips and Jan Villemson, "*E-election Processes on the Example of the Parliamentary Elections of the Republic of Estonia*", Master thesis, University of Tartu, Aug 2024, https://dspace.ut.ee/items/a6aca03c-8dd9-4519-9ae3-d6a873316c76
31. "*E-voting system Disk appeared out of nowhere*", https://gafgaf.infoaed.ee/posts/esoteeriline-turvamudel/; last accessed 22/5/2024.
32. Election Regulative Law in Estonia, https://www.riigiteataja.ee/en/eli/501102024002/consolide#para48b8
33. The incident video, 20/10/2025, https://youtu.be/Ccc7I40OT-o; last accessed 23/10/2025
34. The report about the incident; https://ausadvalimised.ee/docs/kov2025-eelraport/
35. Official Statement, after the video: 23/10/2025, https://www.facebook.com/share/p/17ZDuRjcRe/; before the video: 21/10/2025, https://www.facebook.com/share/p/15ho283vG3/; last accessed 27/10/2025.
36. EKRE request for recount , 22/10/2025: https://www.facebook.com/share/p/1FaF3i5zv1/, last accessed 27/10/2025; 4/11/2025: https://uueduudised.ee/martin-helme-valimiskomisjoni-istung-paljastas-tuima-panevate-ametnike-ringkaitse/, last accessed 6/11/2025.

37. Poll station voter finds a recorded e-vote for him, 20/10/2025, https://x.com/PriitTammik/status/1980352232718795045; last accessed 27/10/2025
38. Shymaa M. Arafat, "*Automated Ballot Stuffing with an Encrypted Vote: A Large-Scale Attack on the Estonian Internet Voting System (IVXV) and its Mitigation*", E-Vote-ID 2025, Track-4 Posters and Demos, 10th International Conference on Electronic Voting, 1-3 Oct 2025, Nancy France.
39. Estonian Academy of Sciences, "*No HIGH security risks were detected in Estonian election technology*", 1 Oct 2024, https://www.akadeemia.ee/eesti-valimiste-tehnoloogias-ei-tuvastatud-korgeid-turvariske/; last accessed 12/11/2025.
40. Arnis Parsovs, "*Estonian Electronic Identity Card: Security Flaws in Key Management*", 29th USENIX Security Symposium, Aug 2020, 978-1-939133-17-5; video https://www.usenix.org/conference/usenixsecurity20/presentation/parsovs; last accessed 24/3/2024.
41. Arnis Parsovs, "*Estonian Electronic Identity Card and its Security Challenges*", 2021, PhD Thesis, University of Tartu.
42. Geenius, *"The police discovered 15,000 faulty ID cards, over 300 have been used (in Estonian)*", June 2019. https://digi.geenius.ee/rubriik/uudis/politsei-avastas-15-000-veaga-id-kaartiule-300-on-kasutatud/; last accessed 20/3/2024.
43. Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, Vashek Matyas, "*The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli*", CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages1631 - 1648, https://dl.acm.org/doi/10.1145/3133956.3133969
44. https://e-estonia.com/raulwalter-estonia-digital-identity-giant/; last accessed 20/3/2024
45. https://e-estonia.com/estonia-introduced-a-new-id-card/; last accessed 20/3/2024.
46. https://valimised.ee/sites/default/files/2023-02/IVXV-protocols.pdf
47. Kristjan Krips, Supervised by Jan Willemson, "*Privacy and Coercion Resistance in Voting*", June 2022, PhD Thesis, University of Tartu, dspace.ut.ee/server/api/core/bitstreams/58ffcbf3-7cc8-4381-b7ca-a9d3e777dcd6/content; last accessed 10/7/2025
48. Jan Willemson, "*Creating a Decryption Proof Verifier for the Estonian Internet Voting System*", ARES 2023, Italy, ACM ISBN 979-8-4007-0772-8/23/08, https://doi.org/10.1145/3600160.3605467
49. Anggrio Sutopo, Thomas Haines, Peter Rønne. "*On the Auditability of the Estonian IVXV System and an Attack on Individual Verifiability*". Workshop on Advances in Secure Electronic Voting, May 2023, Bol, brac, Croatia. https://link.springer.com/chapter/10.1007/978-3-031-48806-1_2
50. Olivier Pereira, "*Individual Verifiability and Revoting in the Estonian Internet Voting System*", 2022, https://www.researchgate.net/publication/372570425_Individual_Verifiability_and_Revoting_in_the_Estonian_Internet_Voting_System
51. S. Heiberg and J. Willemson, "*Verifiable Internet Voting in Estonia*", 2014, 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE), Austria, pages 1-8, https://ieeexplore.ieee.org/document/7001135
52. D. Springall et al., "*Security Analysis of the Estonian Internet Voting System*", In Proceedings of the 2014 ACM SIGSAC Confrence on Computer and Communications Security (CCS'14), pages 703-715, https://dl.acm.org/doi/10.1145/2660267.2660315
53. https://research.cyber.ee/~janwil/publ/ivxv-evoteid.pdf

54. Smartmatic-Cybernetica. IVXV Voting Service. Version 1.8.2-RK2023, https://github.com/valimised/ivxv/tree/master
55. http://www.ria.ee/en/i-voting-20-years-progress; last accessed 23/11/2025.
56. https://github.com/valimised/ivotingverification/blob/published/app/src/main/java/ee/vvk/ivotingverification/util/ElGamalPub.java#L77-L83, and https://github.com/valimised/ios-ivotingverification/blob/published/VVK/Crypto.m#L141-L146; last accessed 18/5/2025 (previous code version)
57. Server-Side part of the Voting Application, first published 30/9/2025, https://github.com/valimised/ivxv/blob/published/voting/service/voting/main.go; last accessed 31/10/2025.
58. Valimised, "*Protocols PDF"*: https://www.valimised.ee/sites/default/files/2024-05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-protokollide%20kirjeldus%29.pdf; "*Architecture PDF*": https://www.valimised.ee/sites/default/files/2024-05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-arhitektuur%29.pdf
59. The *Decrypt tool in the key application*, IVXV 1.9.10 EP2024, https://github.com/valimised/ivxv/blob/published/key/src/main/java/ee/ivxv/key/tool/DecryptTool.java#L247; last accessed 5/7/2025
60. J. Müller, "*Breaking and Fixing Vote Privacy of the Estonian E-Voting Protocol IVXV*", In: Matsuo, S., et al. Financial Cryptography and Data Security. FC 2022 International Workshops. FC 2022. LNCS(13412), https://doi.org/10.1007/978-3-031-32415-4_22
61. Krips, K., Snetkov, N., Vakarjuk, J., Willemson, *"Trust Assumptions in Voting Systems"*. In Computer Security. ESORICS 2023 International Workshops. Lecture Notes in Computer Science, vol 14399. Springer, Cham, https://doi.org/10.1007/978-3-031-54129-2_18; full paper at https://arxiv.org/pdf/2309.10391
62. https://github.com/valimised/ivxv/blob/published/auditor/src/main/java/ee/ivxv/audit/tools/IntegrityTool.java
63. Tarvo Treier and Kristjan Duuna, "*Identifying and Solving a Vulnerability in the Estonian Internet Voting Process: Subverting Ballot Integrity Without Detection*", IEEE Access, Vol.12, https://ieeexplore.ieee.org/document/10811882
64. https://myid.skidsolutions.eu/en; last accessed 13/7/2024.
65. Interaction with IVXV, 3/2024, https://anonymous.4open.science/r/SoK_Estonia_IVXV_EVoteID-C2E4/Letter_to_IVXV_with_replies.pdf
66. Sevdenur Baloglu, Sergiu Bursuc, Sjouke Mauw, and Jun Pang, "*Formal Verification and Solutions for Estonian E-Voting*", In ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 2024, Singapore, Singapore. ACM, New York, NY, USA, https://doi.org/ 10.1145/3634737.3657009
67. Tarvo Treier, "*Re-voting Under Surveillance: National eID Transaction Logs as a Threat to Coercion Resistance in Estonian Internet Voting*", E-Vote-ID 2025, LNCS pp. 191-207, Oct 2025, https://link.springer.com/chapter/10.1007/978-3-032-05036-6_12
68. https://github.com/valimised/ivxv/blob/published/voting/internal/sessionstatus/rpc/client.go (#L22,#L105, #L167); last accessed 3/11/2025 (new version)
69. https://www.valimised.ee/en/internet-voting/guidelines/voter-applications-and-checking-authenticity; last accessed 5/11/2025.
70. Valimised e-voting documents, the Estonian language site with browser translations, files: *IVXV EHS (General Framework)* & *Architecture* & *Protocols*, https://www.valimised.ee/et/e-haaletamine/dokumendid; last accessed 8/11/2025

71. https://github.com/valimised/ivxv/blob/published/common/java/src/main/java/ee/ivxv/common/crypto/CorrectnessUtil.java; last accessed 5/11/2025.
72. https://github.com/valimised/ivxv-mixnet-adapter/commits?author=svenheiberg; last accessed 8/11/2025.
73. Taaniel Kraavi & Jan Willemson, "*Proving vote correctness in the IVXV internet voting system*", Spring Nature Scientific Reports, (2025) 15:31793, https://doi.org/10.1038/s41598-025-16764-1
74. Extended version of the paper, https://anonymous.4open.science/r/SoK_Estonia_IVXV_EVoteID-C2E4/Estonia_EVoteID_long.pdf
75. Vincent Cheval, Charlie Jacomme, Steve Kremer, and Robert Künnemann, "*SAPIC+: Protocol Verifiers of the World, Unite!*", In 31st USENIX Secu rity Symposium, USENIX Security 2022, Boston, MA, USA, August 2022, Kevin R. B. Butler and Kurt Thomas (Eds.). USENIX Association, 3935–3952, https://www.usenix.org/conference/usenixsecurity22/presentation/cheval
76. Shymaa M. Arafat, "*Removing Insiders' Trust from The Estonian Internet Voting System (IVXV)*", E-Vote-ID 2025, Track-4 Posters and Demos, 10th International Conference on Electronic Voting, 1-3 Oct 2025, Nancy France.
77. Valeh Farzaliyev and Jan Willemson, "*End-to-End Verifiable Internet Voting with Partially Private Bulletin Boards*", E-Vote-ID 2025, LNCS pp. 73-89, Oct 2025, https://link.springer.com/chapter/10.1007/978-3-032-05036-6_5
78. Ertem Nusret Tas and Dan Boneh, "*Vector Commitments with Efficient Updates*", arXiv:2307.04085v5 [cs.CR] 4 May 2024
79. B. Blanchet et al, "*ProVerif: Cryptographic Protocol Verifier in the Formal Model*", https://bblanche.gitlabpages.inria.fr/proverif/; last accessed 22/11/2025.
80. J. Dreier et al., "Shaken, not-Stirred – Automated Discovery of Subtle Attacks on Protocols Using Mix-Nets", In 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, USA, August 2024; hal-04615474v1
81. Switzerland Internet Voting System, Swiss Post, https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/tree/master, last accessed 11/12/2024.
82. Public Bulletin Board attacks
83. James Austgen, Andrés Fábrega, Sarah Allen, Kushal Babel, Mahimna Kelkar, Ari Juels, "*DAO Decentralization: Voting-Bloc Entropy, Bribery, and Dark DAOs*"; https://arxiv.org/abs/2311.03530; https://github.com/DAO-Decentralization/dark-dao/tree/main; last accessed 20/3/2024.
84. Grok, "*Batching Electrum-like Checks in 1-Button Click for Authenticating IVXV Voting Application*", 9/7/2025, https://anonymous.4open.science/r/SoK_Estonia_IVXV_EVoteID-C2E4/Grok_X_Electrum_1button_IVXV.pdf
85. Cyber Security Committee 2025' report, https://koodivaramu.eesti.ee/riigi-valimisteenistus/valimised-turve/-/blob/main/2025/kaaskiri-2025.md; last accessed 25/11/2025.
86. Cyber Security Commission minutes of meetings, https://www.akadeemia.ee/akadeemia/noukogud-ja-komisjonid/kuberturvalisuse-komisjon/; last accessed 7/7/2024.
87. Estonian Academy of Sciences, *Conference on Trust and Soundness*, 28 Oct 2024, https://www.akadeemia.ee/sundmused/konverents-usaldusest-ja-usaldatavusest-2024/; last accessed 12/11/2025.

88. Estonian Academy of Sciences, "*The risks associated with elections must be discussed openly*", 16 Oct 2024, https://www.akadeemia.ee/valimistega-seotud-riskidest-tuleb-raakida-avalikult/; last accessed 12/11/2025.

89. https://www.bka.de/DE/Presse/Listenseite_Pressemitteilungen/2022/Presse2022/220405_PM_IllegalerDarknetMarktplatz.html; written 5/4/2022, last accessed 20/7/2025.
90. https://wwwcomputest.nl/en/knowledge-platform/blog/arrests-worldwide-genesis-market-for-online-identity-fraud/; written 5/4/2023, last accessed 20/7/2025.
91. https://www.bitsight.com/blog/what-are-compromised-credentials; written 14/5/2025, last accessed 20/7/2025.
92. https://www.olfeo.com/en/les-serveurs-command-control/; written 29/2/2024, last accessed 20/7/2025.
93. https://www.zdnet.com/article/cybercrime-market-selling-full-digital-fingerprints-of-over-60000-users/; written 9/4/2019, last accessed 20/7/2025.
94. https://www.bleepingcomputer.com/news/security/google-sues-to-disrupt-badbox-20-botnet-infecting-10-million-devices/; written 17/7/2025, last accessed 20/7/2025.
95. https://www.theguardian.com/world/2023/feb/15/revealed-disinformation-team-jorge-claim-meddling-elections-tal-hanan; written 15/2/2023, last accessed 20/7/2025.
96. https://www.eset.com/blog/en/home-topics/cybersecurity-protection/voters-cybercriminals-elections-safety/; written 31/5/2024, last accessed 20/7/2025.
97. T. Niraula et al., "*Quantum Computers' threat on Current Cryptographic Measures and Possible Solutions*", Oct 2022, International Journal of Wireless and Microwave Technologies 12(5):10-20, DOI:10.5815/ijwmt.2022.05.02, https://www.researchgate.net/publication/368394434_Quantum_Computers'_threat_on_Current_Cryptographic_Measures_and_Possible_Solutions
98. https://a16zcrypto.com/posts/podcast/quantum-computing-what-when-where-how-facts-vs-fiction/; last accessed 17/5/2025.
99. A.R. Perez et al., "*An electoral exception? Quantum computing - readiness and internet voting*", JeDEM Issue 16 (3): 50-79, 2024, DOI: 10.29379/jedem.v16i3.928
100. https://www.mkm.ee/en/news/estonia-joined-eus-cooperation-framework-quantum-communication-infrastructure; last accessed 28/4/2025.
101. https://sciencebusiness.net/network-updates/university-tartu-and-cybernetica-cooperate-study-quantum-safe-cryptography; last accessed 28/4/2025.
102. J. Vakarjuk, N. Snetkov, and P. Laud, "*Identifying Obstacles of PQC Migration in E-Estonia*", 2024, https://ieeexplore.ieee.org/document/10685570; full paper available at https://ccdcoe.org/uploads/2024/05/CyCon_2024_Vakarjuk_Snetkov_Laud-1.pdf
103. https://cordis.europa.eu/project/id/101087529; last accessed 28/4/2025.
104. https://www.enisa.europa.eu/sites/default/files/2024-11/Remote%20ID%20Proofing%20Good%20Practices_en_0.pdf; last accessed 22/4/2025.
105. https://e-estonia.com/cybernetica-post-quantum-cryptography-joins-to-menu/; last accessed 28/4/2025.

Appendix A: Handling of the election secret Key

First, recall from section 4.3 that the election secret key ($\mathbf{S_k}$) is only used by IVXV when counting the votes (to allow the decryption of the votes without knowing the voter ID or the random number "r"); hence, the straightforward risk from key leakage concerns ***privacy***. However, let's investigate the possibility of integrity risks.

-When counting votes, the election authority uses the election secret key ($\mathbf{S_k}$) and El-Gamal encryption known equation « $\mathbf{y= g^{Sk}}$ » to calculate $\mathbf{v= C_2/((C_1)^{Sk})}$
-What prevents an adversary from stuffing ballot with any dummy pairs (C_1,C_2) such that $\mathbf{C_2/((C_1)^{Sk}) = v_{adversary}}$**?**

1. An outsider adversary cannot do that without compromising voters' credentials to add a valid signature; otherwise, the Vote Collector (VC) will detect it when contacting the Registration Service (RS) defining the voter as "not eligible to vote". Hence, these kinds of attacks do not really need the election secret key and fall into malicious voting applications and absent voting discussed in section.

2. An insider adversary can do that only, even in the newer versions, if he/she was able to compromise both VC & RS to fake the certificates and pass their integrity checks; however, if the Ballot Processor (BP) performs a cryptographic check on the online count of votes it can detect a change in that count.

Appendix B: Summarizing Tables

Table.1: Fixes/improvements done in IVXV 1.9.10 EP2024, 1.10 E2025

Risk	Deployed Solution	Remaining Issues	GiHub File	Corresponding Academic Research
Invalid votes	**2024:** Decrypted in a separate file with ZKPs of correct decryption	-Files are viewable by auditors only (complaints)	Embedded in [60] ***DecryptTool.java***	Tallinn Univ. Ms. thesis [19] (Jun 2024)
	2025: Deployed ***Range Proofs*** to prevent invalid votes from entering the votes list.	-Invalid votes (3 in 2025) can still enter the system through private voting applications (through the VC directly); hence, privacy attacks [61] remains.	Added in the server-side part of the voting application [58]	The thesis in [19] and the scientific report in [74]
The Voting/Voter Application (VA)	**2024:** Authenticated through a fingerprint [70]	-Complicated and involves many steps for regular users		Jan Willemson, E-Vote-ID 2025 paper [9]
	2025: Publishing the server-side code of the voting application [58]	Assumes trust in the official VA (client- side part) -Problems remains for non-verifying and absent voters (see Table.2)	[58] https://github.com/valimised/ivxv/blob/published/voting/service/voting/main.go	
Ballot Processor (BP) manipulation	**2024:** Consistency checks on SHA256 hashes of totals and subtotals.	***-Offline*** checks; i.e., count based validation depends on trusting the Vote Collector (VC) and Registration Service (RS) to not collude before the list enters the BP.	A new file [63] ***IntegrityTool.java***	Tallinn Univ. researchers [64] (Dec 2024)
Timing attacks	**2024:** Checking *Session ID* and *Timestamps* difference, which are generated by *PKIX* protocol	Cannot detect fast attacks that can manage to work in the duration of one session (like ***Pereira attack*** [51])	A new file [50] ***client.go***	An extension, [67], to a Luxemburg Univ. PhD on formal verification of i-voting systems, applied to IVXV (Jun 2024)

Table.2: Remaining vulnerabilities/risks in IVXV 1.10 E2025 and suggested solutions

Vulnerability	Risks/threats	Concerns/Complaints about the issue	Suggested Solutions	Proposed by
The Voting Application (VA)	-Pereira attack [51] -Copy attack on Privacy [67] -Large-scale vote copying [39] -Variety of malicious VA risks -Extra risks for smart phone VA -Injecting votes through trojan horses embedded in apps that get the voter signature claiming other purposes; risky even for verifying voters.	-Cybernetica supervised PhD [48/sec.5-6] -Olivier Pereira [51] -OSCE/ODHIR 2023 report [2] -Many other researchers including the authors of this paper.	Using a ***microcontroller*** voting device	Tallinn Univ. PhD [47] (2022)
			-Publishing the VA code (change the law) -Batched easier checking of ***file hash***, [85] -Assigning a ***signature key*** for VA and allowing ***optional registering of other VAs*** but after scanning the code for malicious activities (more robust but require flexibility and cooperation from authorities to not reject unobjectively). -Activate an ***SMS ack*** with every election transaction on election days; could be delayed as discussed in [51] or discovered by vote coercers [9,17,68]. -Let the Ballot Processor, section 5.2, check and ***alarm the existence of equal encryptions*** even from different voters. -If the system insists on keeping the VA situation as is, using ***a separate ID card for elections*** (suggested in [68] for another reason) can protect from the possibility of casting votes through apps with *trojan horse* that take the voter's signature claiming other purposes. Another solution is to ***add a meta data field*** that allows IVXV to check ***the application category***.	Poster [39] in E-Vote-ID 2025, and this paper
Insiders' Trust	VC and RS are trusted to not collude; their collusion may result in: -privacy attacks [61,67] -different possible manipulations of the ballots list before entering the ballot processor	-Estonian parties and i-voting opposing communities in general [1,2, 16] -Detected by automated formal verification tools in [67,81].	-Adding a ***ZKP*** to each vote.	[61] (2022)
			-Performing different ***consistency check ZK queries***, and ***RLA***s, between ballots list and other services recording digital transactions in Estonia, like *myID* [65]. -Using ***Verkle Trees*** based on KZG commitments to cryptographically prove votes integrity and counters.	E-Vote-ID 2025 poster [77]
			-Using ***Verkle Trees*** (***Vector Commitments***) based on Pedersen commitments to cryptographically prove vote correctness.	E-Vote-ID 2025 paper [78]
Integrity vs. Coercion	Double checking services like myID [65] can be used by vote' coercers to prove last vote.	-Observer's E-Vote-ID poster in 2023 [17]	-Making election transactions secret or obfuscated -Using a special ID card for e-voting that doesn't allow such integrity checks on its transactions.	E-Vote-ID 2025 paper [68]
Absent Voters	-If their devices and credentials are compromised as *botnets* or through any dark web market, votes can be injected on their behalf. -This could happen through trojan horse apps	-Falls under "*unavoidable risks that can't be performed on large-scale"* by [48, 48/ref.166] as it will cause "*observable anomalies*" -Falls under (*corrupted voter device+ corrupted communication network*)	<u>Only safeguards</u>, no complete protection (Appendix C): -Activate an ***SMS ack*** with every election transaction on election days; could be delayed as discussed in [51] or discovered by vote coercers [9,17,68]. -use a SNARK that supports ***Non-inclusion proofs***, and check RLA samples; voters could lie to falsify elections. -Allow a ***"reject all" choice*** to incentivize even	This paper

	too.	category detected in [67]	boycotters to vote	
Poll Station manipulation	A malicious poll station worker could alter vote IDs (keeping the number consistent with the ballot box count) to delete the e-votes of some (by faking a booth vote) and double the votes of others (by giving them the chance to add another online vote)	To our knowledge, the Estonian Cyber Security Academy report, [86], was the first to identify the vulnerability as a "*noteworthy*" threat.	-Notify the voter through an email or any other notification channel of a poll station vote; could be deliberately delayed as [51] worries.	Cyber Security Academy report [86]
			-Check for anomalies like an e-vote with time stamp larger than the timing of the corresponding poll station vote. -Take a random sample of poll station cancelled votes and contact them to check the truth before deleting; then apply RLA techniques to decide on the result.	This paper
Quantum Computing	-Breaks existing cryptography -Privacy Risks (HNDL)	-The paper [102] (2024) -HNDL in earlier versions of this paper [75] (2025), and Cyber Security Committee report [86] (2025)	-Prevent data leakage -Prepare carefully without rush the transition to post quantum cryptography (PQC)	
AI	-Leakage of voter's vote -Organize a sophisticated attack	-The leakage in the [86] report as a less significant threat. -The attack orchestrating threat is warned by this paper.	None yet	

Appendix C: Absent Voting

The threat of online identity fraud through dark web markets [89,90] that sell compromised users' credentials [91], or devices to be used as botnets [92], has been widely acknowledged long ago, since 2019 [93], and continues to be [94]. While the stories published, like [95,96], talk mainly about affecting public opinion during elections, the technologies used do not exclude at all the possibility of online voting using stolen credentials.

The Swiss online voting system, SwissPost [82], guarantees integrity even with compromised voter device through using traditional paper mail as another channel; voters assert their personhood identity by entering codes they received by regular mail. In Estonia, IVXV depends on vote verification as the safeguard from such attacks which are kind of unavoidable risks on internet voting. Also, researchers like [48, 48/ref.166] counted on observable anomalies (like financial theft if the digital ID is compromised) as a guarantee that those kinds of attacks cannot go unnoticed if performed on a large-scale.

While we discussed how authenticating the voting application would provide more security guarantees even for nonverifying voters, we stopped to think that there are more risks associated with those who did not vote at all (no ballot to generate a QR-code or ZKP from). We believe counting on observable anomalies is not enough here; an adversary may buy compromised credentials only for political purposes, at least till the election ends and then uses them for stealing money. Also, equally likely, is the possibility of deceiving even verifying voters when using their eID in any other online activity (requiring a signature) after voting; a malicious application could contain a trojan horse that absently votes for whomever uses it.

With the current existing cyber security threats, there is no solid guarantee for those who chose not to vote that no one stole their voting credentials and voted instead of them. However, we suggest the following partial protections:

- Provide a "reject all" choice in every list of candidates to encourage boycotters to vote anyway and hence exclude this from this risk category.
- Educate voters about online identity fraud and check discovered criminal markets for Estonian victims.
- A separate election ID can only prevent absent voting through trojan horses.
- Activate an SMS acknowledgement for every digital ID transaction (like some banks provide for credit cards transactions) on election days. This way, an IVXV transaction SMS for someone who didn't vote will alarm the voter to take an action; for example, overwrite the vote whether electronically or in poll stations. Still, this solution is subject to deliberate message delays as discussed by Olivier Pereira in [51].
- If the used ZKP supports non-inclusion proofs, like Verkle Trees42, a random sample from non-voters could be taken and checked for non-inclusion proofs in an RLA manner. This solution has a drawback of not being able to distinguish liars; i.e., if the RLA showed a large ratio of absent voting, how would a court know beyond reasonable doubt that those people are not deliberately lying (did vote by themselves) to falsify the election.

Appendix D: The Quantum Computing (QC) threat

Quantum computing depends on physics and quantum theory to perform computations much faster than current existing hardware in a way that can break most existing cryptographic systems, [97] like those based on the discrete logarithm problem (El-Gamal encryption) or Elliptic Curve Cryptography; hence, it is a threat to IVXV and the Cyber Security Committee Sep 2025 report, [86/noteworthy threats/quantum], acknowledges it.

Most digital information systems around the globe have started their research for alternatives that can defeat the quantum computing threat, what is known as ***Post***

[42] Note that in this case, there should be another Verkle (or any ZKP) for voteIDs only; ie, not for the complete ballot as non-voters do not have ballots at all.

Quantum Cryptography (PQC). Also, if the threat is not that approaching, there is an associated with another risk of storing encrypted data to decrypt them later when QC is possible [98]; the concept of ***Harvest Now Decrypt Later (HNDL)*** can jeopardize vote privacy[43] and necessitates a firm restriction on the leakage of election encrypted data before the usual timed destruction.

Estonian Quantum Efforts

All countries deploying e-voting/e-government systems should be motivated, without rushing unstudied, to provide post-quantum solutions. For our specific case of Estonia, the authors of [99] included 5 out of 24 experts from Estonia in their interviews, and covered steps taken by the Estonian government [99/ sec. 2.2.2] in different quantum related aspects.

-A European cooperation to deploy a quantum communication infrastructure in 2020 led to the NordIQuEst (*Nordic-Estonian Quantum Computing e-Infrastructure Quest*) in 2022 [100]

-A collaboration between *Cybernetica* and University of Tartu to create post quantum solutions has started in 2021 [101] including jointly supervised doctoral thesis from which we encountered a paper (2024) on PQC migration obstacles in Estonia [102].

-Finally, PQC is 1 of 6 challenge areas included the *Cyber Security hub* established in 2023 between Estonia and a major Czech ICT powerhouse both in industry and education [103].

Estonian I-Voting Related Efforts

Digital IDs are a crucial element in i-voting; the *ENISA* July 2024 report [104] on the unified European digital identity project *eIDAS*, though, did not specify the PQC transition details. Then on 19 Feb 2025, [105], *Cybernetica* announced that the first hardware chip enabling post-quantum ID card has been certified in Europe. When asked about post quantum i-voting, they answered "*there is more math to do*". In their 2024 paper [102/sec. 2.E], they considered the complexity of managing multiple layers a disadvantage in most existing Hybrid systems. For i-voting, [102/sec. 4.E], this would be a problem with the needed multiple layers to shuffle votes in *mix-nets.* As can be concluded from the latest available report, [86], the research work is predictably going to take years.

[43] The report in [86] exact words about *HNDL* from Google translation "*Reorganizing the elections will not help, because people's preferences have become public. This would most likely mean the end of i-voting in Estonia*". Protections include ways/measures to make sure of the destruction of all copies of encrypted votes after the elections