Cryptanalysis of the Estonian i-voting System (IVXV)

Vulnerabilities Associated with the Voter Application

The Estonian i-voting experience is probably the richest to analyze; yet it still causes society division and holds a collection of *noteworthy threats* as stated in the Estonian Academy of Science (Cyber Security Committee) report. This paper is mainly concerned with the Voting/Voter Application (VA) that voters download and vote through; we believe, the Estonian approach in handling the VA introduces many exploits that could affect election integrity. The voting application is not authenticated by the voting system, and its source code is not published. With a vote verification ratio below 10%, not publishing the source code opens the door for doubts on possible insider manipulation, including absent voting, of ~ 90% unverified votes. The argument that allowing voters to use other voting applications they trust compensates the doubt has many disastrous consequences; it provides an open challenge for adversaries to find exploits in other open-sourced parts of the system and invoke them through a malicious VA that impersonates the official one. Furthermore, we introduce in this paper a more severe risk that, to our knowledge, has never been discussed before; any other purpose application that takes the voter signature could be a ***voting trojan horse***. Since the Estonian i-voting system, IVXV, allows multiple voting, trojan horses can overwrite even the votes of voters who followed the tedious VA' authentication steps through Windows/Linux and also verified their votes. With stolen credentials and/or embedded trojan horses, large scale attacks are pretty feasible; this paper discusses risks associated with IVXV handling of the voting application and suggests some solutions. Finally, we close with a whole picture concluding remarks and general questions.

Additional Keywords and Phrases: IVXV, El-Gamal Encryption, verifiability, trojan horses, vote buying/coercion, absent voting.

1 PRECFACE

To understand a system's security requirements, it is important to understand its working environment; the existing conflicts of interest, the possible adversaries/attackers and their estimated power. This section briefs the reader about the Estonian political environment both locally and as part of the European Union (the system was also used for voting in the 2024 European Parliament election).

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[1] [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; the official site [3] shows the ratio of i-votes fluctuates between (46±5)% in the last few years. 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, [4], showing the distribution of internet votes to be completely different from paper votes curves according to each party's preference. Still, there were some complaining activities that continued persisting to following elections; On 10th Nov 2025, the party called for a parliament vote to urgently suspend e-voting, [5,6], that gained 25 approving votes and only 1 refusing vote, while the rest of 87 present members chose not to vote. In fact, we believe the distribution of members attitude reflects more society division that the statistics in [3]; even if the majority (61 out of 87 attending members and 101 total members) did not support "*urgent*" suspension, they did not vote against it for us to classify them as supporting e-voting.

[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.

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 [7] the ongoing war has put extra pressure on the country. The i-voting[2] system faced Russian attacks that authorities say were properly defended; on the other hand, Russia decided to abandon the use of e-voting, [8], during war time. 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 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 exactly their adversaries' wish.

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 explains the current version of the Estonian i-voting system (IVXV) ending with an important attack that was fixed before the 2023 elections. Then, Section 3 is dedicated to the problematic issues in its Voter (sometimes called Voting) application and suggested solutions. Finally, section 4 gives concluding remarks on the system status as whole.

2 IVXV

In this section, we explain the design and structure of the Estonian internet voting system, IVXV, as described in the official documents [10]. Then we detail an important cryptographic attack along with its fix (before the 2023 elections) that will help in demonstrating how the system works.

2.1 Brief Factsheet

The developing companies are *Cybernetica-Smartmatic* [11]; the voting device had to be a desktop PC, smartphone voting is about to become possible; voting can be done using mobile-ID, Smart-ID, or any digital identity integrated in the web-eID[4] ; 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 Mix-net server** is used to scramble votes before decryption to preserve ballot secrecy; an auditor application (could be run by anyone) that verifies the cryptographic **NIZKP proofs** for the mix-net output provided by IVXV. The setup phase, before the election starts, constructs the election secret key from 9 parts issued by the members of the Election Commission of the Republic, such that decryption requires 5 out of 9 key parts using **threshold cryptography**.

2.2 System Architecture & Voting Steps

The system architecture and voting steps are illustrated in Figure.1 and could be summarized as follows

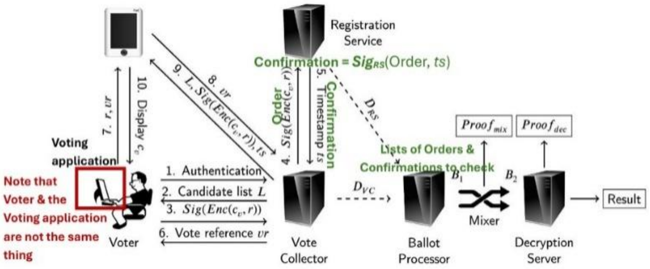

Figure 1: A diagram describing the architecture & the steps of IVXV; adopted from [1] with remarks in red and following updates in green.

1. The voter installs the voting application, sometimes called voter application, and 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*

[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)

[4] 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 [10]). 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

voting application has the right to sign a message with the voter signature), adds also the signed *timestamp certificate*[5] received from the registration application through the vote collector *after verifying the signatures of both*, and then sends the double envelope ballot to IVXV.

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)[6] 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.
5. The vote collector sends a verifying ***QR code***[7] to the voter for optional vote verifying (through verification application) using a second smart device.
6. In the newest October 2025 version, the encrypted ballot in step 4 is first received by a server-side part of VA, [13], which first checks that the vote is valid using ***Bullet Proofs-Pedersen Commitments*** based Range Proofs [14,15] before sending it to the vote collector. However, the system allows non-official VAs that could send their ballots directly to the vote collector; hence, another ballot validity check is added to the verification code [16].

2.3 Cryptographic details of an important attack (fixed in Feb 2023)

The exploit introduced in 2022 by [17], 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 "$\mathbf{y}$" with corresponding secret key "$\mathbf{S_k}$", and "$\mathbf{g}$" be the generator for El-Gamal encryption over the finite field **G**; hence, the equation « $\mathbf{y = g^{Sk}}$ »[8] holds.

-To encrypt a vote "$\mathbf{v}$" the voting application generates a random number "$\mathbf{r}$", so that the encrypted vote is $\mathbf{(C_1, C_2) = (g^r ,\ y^r\ v)}$

-The verification application, working instantly within 30 mins, receives "$\mathbf{r}$" 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 "$\mathbf{v}$" is the same "$\mathbf{v}$" 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 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 $\boldsymbol{y^r v = y^{r'} v'}$ 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* $\boldsymbol{C_1= g^r}$ which was done on Feb 2023, before March 2023 elections. The check methodology has changed in the 2025 version to checking that the pair (C_1,C_2) represents a valid point on the election elliptic curve[9]; for the attack described here, 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***).

3 THE VOTER/VOTING APPLICATION (VA)

The main debatable problematic issues about the voting application in the Estonian i-voting system circulates around authentication and open sourcing. We first summarize previous related work and election incidents. Then delve into possible threats and suggest solutions.

3.1 Related Work

3.1.1 Open Sourcing the VA code

In 2022, the authors found it concerning [17/sec. 3.6] that such a straightforward vulnerability like the one described in section 2.3 wasn't noticed earlier; they also criticized IVXV in general [17/sec. 4] especially in keeping the source code of the voting

[5] 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, *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*.

[6] IVXV uses *Douglas Wikström's Verificatum* (https://www.verificatum.org/); the package provides a verification application, and IVXV too (and several other projects [12]).

[7] According to [4], 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.

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

[9] Since 30/90/2025, the verification code 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.

application secret. It turned out that the Estonian e-voting regulation law [18/§48^{8}/(7)] clearly states that "*The source code for the voter application is not published*"; in fact, this contradicts with a newer update [18/§48^{3}/(7)] which states: "*Prior to the start of electronic voting, the State Electoral Office publishes the voter application,…*"]. Apparently, publishing the code of the server-side part of the voter application [13] was the compromise they made to get away with such contradiction. The PhD in [19] 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.

The most recent report from the Cyber Security Committee of the Estonian Academy of Science, [20], studied the debatable threats of both open and closed source code voter application (VA) and considered their impact 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. However, 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; there have been voting incidents with other voting applications, and the report acknowledges this fact in other parts.

3.1.2 Authenticating the Voting Application (VA)

Unlike the rest of IVXV entities in Figure 1, the VA does not have a signature key and is not authenticated by the system. Probably, the designers and/or regulators philosophy is to compensate for hiding the official VA code; whoever doubts can write another one and vote with it. In 2023, an election observer did write his own VA, [21], and voted with it; with the help of other activists, he keep updating and broadcasting it with each election [22]. The OSCE/ODIHR 2023 report, [2/page 8], believes that not authenticating the VA "*could present a cyber security risk*"; implicitly, the Estonian Academy of Science report in [20] agrees with that when it mentions the risk of *rogue applications*. The fact is it is an exploit left for adversaries as an open challenge to make their best shot in taking advantage of it.

On the academic research side, the old verification vulnerability discovered by [17] and all other risks introduced by previous literature whether associated with a corrupted voter device, or a corrupted communication channel, are only feasible with a malicious voting application; researchers in [23/Table1] enumerated different possibilities through applying automated formal verification, and the extended version of this paper [24] contains extensive details according to the most updated IVXV version of Oct 2025. A specific attack that was discovered by Olivier Pereiera, [25], received a lot of attention; a malicious VA could fake a system crash to deceive the voter to vote again, and hence take the voter signature twice (generate another "***r***" value to construct and sign a new encrypted vote in the background). The malicious VA in this case can show the voter the QR code of his/her choice, while it will be considered an old vote by the system and will be overridden by the adversary choice; [23] notified that the attacker can delay casting the new vote to the last possible time to guarantee it is the last vote, what they called a "***timing attack***". From different solutions, [24/sec 5.3], suggested by the original paper [25] and other researchers, IVXV acted mainly by checking the session ID of the encrypted ballot has not changed before generating the QR-code of the vote; the code documentation [26/lines 22&105&167][10] highlights that this "*prevents reusing session ID until it is deleted from a database or expired*".

The session ID solution handles any kind of timing attacks including [26], but not the general problem of a malicious voting application. Although it does not score well in usability, and no available statistics of its usage ratio[11], IVXV added a possible check for voters to authenticate the official VA after download [27]. The researcher in [9/sec.4], as one of IVXV team, argues that if voters reported crashes with probability "***p***", a large-scale attack is infeasible because it will be detected with probability **$(1- p)^{n}$** where ***n*** is the scale of the attack; the author also argues that an adversary trying to fake a developer key for a malicious voting application will leave traces, which we doubt is definite with overwhelming probability [28] for a nation scale adversary as possible in European parliament elections for example. Hence, we believe that the voter application (VA) is the most serious integrity threat of IVXV, and the Estonian Science of Academy report seems to agree with us [20, 24/sec. 7.5.1] in a sense; 5 out of its 9 *noteworthy threats* are concerned with the voting application.

[10] The session ID check was added in 2024; 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, [13], and almost every code that deals with the vote.

[11] The most recent study on a random sample of voters (https://academic.oup.com/iwc/advance-article/doi/10.1093/iwc/iwaf055/8407037) did not measure that, and we tried to contact the author to see whether such a question was asked in any intermediate unpublished results but couldn't find an answer.

3.2 Risks associated with VA

Besides the usability issues regarding the available VA authentication method [27], the official VA itself is a problematic trust issue as long as the code is not published. The QR code verification is an enough solution for ***individual verifiability***[12], but with more than 90% non-verifying voters [3] it is not enough evidence for ***universal verifiability***; election integrity is under so many risks; if non-verifying voters were identified, for example through social engineering, an insider who has enough access rights to the official website can make them download a malicious modified VA copy. Furthermore, ballot stuffing with absent votes could be a reason behind this high ratio; ***absent voting is possible*** using stolen voters' credentials, [24/Appendix C], or through trojan horse voting applications that claims other purpose.

In addition to the open challenge for adversaries to design a malicious VA that invokes any hidden undiscovered exploits in the system; a more severe possibility is ***a trojan horse*** *application that deceives the voter to take his/her signature for another purpose* like online account registration; Figure 2. A voter who authenticated the VA before voting and verified the QR code after voting, can still be tricked to download a malicious application offering a highly discounted item or a tempting profitable trading deal and its account registration[13] is signed using the same digital ID; the voter has no way of detecting if such an application contains a malicious code that casts a vote using the given signature. The threat could also come from an already installed legitimate application (whom vendor either has a secret political bias or has been bribed to do so); the application may perform an automatic update on election day to include the trojan then updates again after the election ends to remove it.

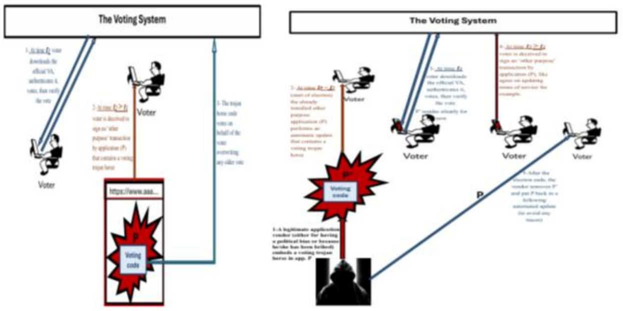

Figure 2: two cases for applications with a voting trojan horse

Moreover, *new technologies like online coding through blockchains and smart contracts*, [29,30,31], have introduced more capabilities to render automated large-scale[14] attacks, with online auctions and even execution attests for vote buyers/coercers, quite feasible. The attack in [32], which combines the *copy attack* in [23] with techniques in [29,30,31], is a real existing threat. Voters and even intermediate colliders may not know who they are voting for, they just receive an encrypted ballot from the adversary $\mathbf{(g^r, y^r\ v)}$ and add a fresh randomness to it $\mathbf{(g^{r+rnew},\ y^{r+new}\ v)}$; the encrypted ballot could be sent by another country, or could be determined online through an online auction between malicious candidates. Note that this attack can be performed using trojan horse applications as well; as for cost, the complete attack was implemented in [29] for blockchain DAO (Decentralized Autonomous Organization) voting, and yet it can be implemented much easier and less costly with an adversary (possibly foreign) sending an encrypted ballot to an intermediary.

3.3 Suggested Solutions

-The only defenses we find possible from the trojan horse case, if the system insists on not authenticating the VA, is for the Estonian government to separate i-voting transactions. This could happen **by adding a meta data field that identifies the application category** so that IVXV does not accept transactions from applications registered as gaming, buying,...etc; however, this solution involves the e-government system to accept it and apply it. Another way to separate e-voting, is the suggestion in [33] of **using a separate digital ID for e-voting transactions**, though suggested for another purpose; Still, this does not comply with the system policy of allowing more digital IDs (footnote 4).

-A safeguard protection from the copy attack in [23] is **to check for equal values in the list of encrypted votes** and raise an alarm if found any. This, happened in 2025 [24/sec. 2.3.2], 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 [32] if no fresh randomness is added to each clone**.**

-Other suggested solutions are mostly straight forward; change the law to **publish the official voting application**; provide a

[12] 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). ***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 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).

[13] Although buying transaction are usually signed using bank cards, there are a variety of possible applications which their initial registration (account sign up) may involve signing using one of the digital IDs allowed in the election.

[14] Previous researchers like [19] (based on [19/ref.166]) counted on observable anomalies (ex.: 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 adversaries that might be interested only in altering election results, and others who may defer any financial malicious activities till the election ends. Besides, [19/ref.166] was written in 2014 before smart contracts provided online execution [29/sec.3&5], and TEEs (trusted Execution Environments) provided execution attests. Researchers warned from possible "evil" uses in e-voting ([30] since 2019, [29/ref.33] in 2018); their warnings seems not to reach the e-voting research community though.

more **simplified batched authentication** [34]; 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 encourage **nonvoters to check if someone voted for them** using services like *myID* (https://myid.skidsolutions.eu/en). -For the risk of absent voting, [24/Appendix C], we suggest a somewhat new approach to IVXV; observers can take ***a* random sample** from those who did not vote **to be checked against the election results** using traditional known statistical methods like different *Risk Limiting Audits* ***(RLA)*** techniques.

4 CONCLUDING REMARKS

Designing a robust i-voting system that lasts for 20 years is not an easy task in anyway and the Estonian i-voting system (IVXV) is continuously evolving and improving with time; we traced many improvements in IVXV 1.9.10-EP2024 and then in 1.10-E2025[15] that are detailed in the extended version [24/secs 5&6]. Remaining vulnerabilities/threats that has been discussed by the literature and/or election observers, [24/sec.7&Appendix B/Table2], include the handling of key parts in the setup phase, adding a Non-interactive Zero Knowledge Proof (NIZKP) to each ballot, adding more mix-net server, integrity-coercion trade off, correct poll station vote cancelling, handling forth coming AI and Quantum Computing threats; last but not least, the problems associated with its voting application, which is the main concern of this paper.

The IVXV voter application has two main persisting issues; not being authenticated properly and not being published as an open source. After introducing the voting system and its working environment, the paper investigated available material and research work on the voter application. We emphasized the possible impacts of the current situation, alarming of new feasible risks that have not been discussed before; then, we suggested some solutions to overcome them and/or solve the voter application problems in general.

Finally, as a closure, we introduce two debatable research questions. The first is ***to what limit can the information provided by general purpose activity logs of digital identities or KYC*** (as much as they are beneficial to ***consistency checks***, as when we suggested their usage in RLAs) ***jeopardize privacy*** (for example helping vote buyers/coercers, [33], in catching voters who try to deceive them) or even worse reveal enough information to impersonate a victim (for example being able to deceive *Complete Knowledge* (*CK*) checks [29/ref.51]). The second is ***whether e-voting systems are classified as a political national security matter that countries ought to be discreet about, or a security research area and a cryptanalysis topic open for researchers.*** The published Estonian academic statements welcome open scientific discussions [35]; however, in reality, the last English language PDF files are dated to the 2022 version[16] and we had to use browsers translators, trace GitHub commits, social media posts, use AI[17] and contact involved personnel.

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/ODIHR 2023 report on Estonian Internet Voting, https://osce.org/files/f/documents/f/f/551179_0.pdf

[3] 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.

[4] https://gafgaf.infoaed.ee/en/posts/perils-of-electronic-voting/; last accessed 4/1/2024.

[5] The Estonian Parliament Voting for "Urgent suspend of e-voting", https://www.riigikogu.ee/tegevus/tooulevaade/haaletused/haaletustulemused-kohalolekukontroll/42bf2e09-0c16-4e1c-b227-478fdb5ad4b9/; last accessed 27/10/2025.

[6] The Estonian Parliament Voting for "Urgent suspend of e-voting", the EKRE party account, https://www.facebook.com/share/p/16cvJgpr9y/; last accessed 27/10/2025

[7] Alexander Martin, 13/3/2023, https://therecord.media/estonia-cyberattack-parliamentary-elections, last accessed 8/7/2025.

[8] REM, 29/9/2025, https://epde.org/?news=russian-authorities-shelve-plans-for-mass-rollout-of-electronics-voting; last accessed 8/2/2026.

[9] 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

[10] 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-

[15] 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 [10], 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, [10/protocols/sec.6]; the interaction between online (RIA), offline (RVT) and other IVXV services. The same applies for the 2025 version.

[16] 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 [24/sec. 7.3], especially that the cited documentation of *Verifactum* in [12] uses a 3-servers example for demonstration.

[17] Through this paper, we used AI in: double checking if there is a type metadata field in the Estonian e-government system, creating a command batch file to authenticate VA in 1-click ([34]), tracing some parts of the huge IVXV code for updates (especially when a version is released only days before a submission deadline for the extended version), checking the traces in [9] since it is beyond our area of expertise ([28]), through which grok informed us of the authentication option in [27] that [9] neglected to mention (added in May 2024 after the OSCE/ODIHR report by the way) , and finally creating/predicting several review reports to refine the paper and detect any weakness points before final submission.

05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-arhitektuur%29.pdf

[11] Smartmatic-Cybernetica. IVXV Voting Service. Version 1.8.2-RK2023, https://github.com/valimised/ivxv/tree/master

[12] 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

[13] 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.

[14] Taaniel Kraavi, supervised by Jan Willemson, “*Proving Vote Correctness in the Estonian Internet Voting System*”, Master thesis, Tallinn University, June 2024,https://digikogu.taltech.ee/et/Download/ffdf0de1e58d455ba3d484400c9123fc.pdf

[15] 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

[16] Ballot validity check in verification code, https://github.com/valimised/ivotingverification/blob/published/app/src/main/java/ee/vvk/ivotingverification/util/CheckBallot.java; last accessed 8/2/2026

[17] 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

[18] Election Regulative Law of Estonia, https://www.riigiteataja.ee/en/eli/501102024002/consolide#para48b8; last accessed 27/11/2025.

[19] 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

[20] Cyber Security Committee 2025’ report, https://koodivaramu.eesti.ee/riigi-valimisteenistus/valimised-turve/-/blob/main/2025/kaaskiri-2025.md; last accessed 25/11/2025

[21] "*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.

[22] Mar Poder et al., “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.

[23] 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

[24] Extended version, https://anonymous.4open.science/r/anonymous_citation-E37B/Estonia_Extended.pdf

[25] Olivier Pereira, “*Individual Verifiability and Revoting in the Estonian Internet Voting System*”, FC’22, https://doi.org/10.1007/978-3-031-32415-4_21

[26] https://github.com/valimised/ivxv/blob/published/voting/internal/sessionstatus/rpc/client.go (#L22,#L105, #L167); last accessed 3/11/2025 (new version)

[27] https://www.valimised.ee/en/internet-voting/guidelines/voter-applications-and-checking-authenticity; last accessed 5/11/2025

[28] Conversation with AI on detecting traces, https://anonymous.4open.science/r/anonymous_citation-E37B/Copilot_traces.pdf

[29] 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

[30] Ivan Puddu, Daniele Lain, Moritz Schneider, Elizaveta Tretiakova, Sinisa Matetic and Srdjan Capkun, “*TEEvil: Identity Lease via Trusted Execution Environments*”, ETH Zurich, May 2019, https://arxiv.org/abs/1903.00449

[31] James Austgen, Andrés Fábrega, Mahimna Kelkar, Dani Vilardell, Sarah Allen, Kushal Babel, Jay Yu and Ari Juels, “*Liquefaction: Privately Liquefying Blockchain Assets*”, SBC’25, https://arxiv.org/abs/2412.02634, https://www.youtube.com/live/84tBYn6qot8?t=3555

[32] Author, “*Automated Ballot Stuffing with an Encrypted Vote: A Large-Scale Attack on the Estonian Internet Voting System (IVXV) and its Mitigation*”, Conference Poster, 2025, https://anonymous.4open.science/r/anonymous_citation-E37B/EncryptedcopyattackPoster_anonymous.pdf

[33] 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

[34] Grok, “*Batching Electrum-like Checks in 1-Button Click for Authenticating IVXV Voting Application*”, 9/7/2025, https://anonymous.4open.science/r/anonymous_citation-E37B/Grok_X_Electrum_1button_IVXV.pdf

[35] 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.