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

Abstract. The Estonian i-voting experience is probably the richest to analyze; a country that is considered a pioneer in digitizing both the government and private sector since 2001, and hence digital voting in 2005, yet there are still some complaints submitted, critics and remarks to consider about the IVXV system. In this paper, we introduce a Systemization of Knowledge of the Estonian IVXV i-voting system and propose some added security enhancements. The presented SoK includes applications implemented by election observers in 2023 & 2024 elections, which, to our knowledge, has never been mentioned and/or analyzed in the academia before. The paper also updates the general knowledge about an extra right given to auditors (but not observers) in the June 2024 European election, recent improvements, and recent complaints. Finally, we discuss the current system status in 2024 EP elections, propose our own suggestions to some remaining vulnerabilities, then raise the inevitable question of the approaching quantum threat.
Keywords: IVXV, El-Gamal Encryption, Verkle Trees, vote buying, counted-as-casted.

1 Introduction

Estonia is a small 1.35m 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 general *digital transition* in 2001[2]; 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*) June **2023** report [2]. This is reflected clearly in the i-voting statistics; although it is the most advanced [3], the official site in [4] shows the ratio of i-votes to total votes to reach its maximum of 51% in 2023 (local Parliament) then down to 41.7% in 2024 (European Parliament). One can trace a long history of objection incidences from certain parties in [1] and [2 page8 footnotes16&17]; the situation was emphasized in 2023 when internet votes flipped the results for one of those parties. Analysts view it as a natural

[1] The live number in 30 April 2025 is 1,347,056 from (https://www.worldometers.info/world-population/estonia-population/). More detailed statistics, but dated to Aug 2024, can be found in (https://www.stat.ee/en/find-statistics/statistics-theme/population/population-figure#); **1,127,312 "citizens"** 296,268 of which **~ 26.28% are from Russian ethnics** and the rest of the population (~ 1.35m-1.127m) are ≥ 1 yr residents.

[2] Although the science direct article [1] mark 2001 as the start of the digital transition (e-government) in Estonia, we notify (based on a previous reviewer objection) that Estonia has databases in their government since the 1990's.

reflection of the society division mentioned above; it is expected for the curves, [5], showing the distribution of internet votes to be completely different than that for poll station paper votes. Still, there were some complaining movements that continued persisting to the 2024 European elections [6].

Since the above references illustrate that most rejections came from right parties, it is appropriate to mention that Ukraine is Estonia's closest neighbor and that according to [7] there were Russian attacks on the i-voting system, but the authorities say it was properly defended[3].

Being aware of 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. Then section 4 explains in detail the current version of IVXV ending with an important attack that was fixed before the 2023 elections, and section 5 goes through recent improvements that were made in IVXV before the European elections. Section 6 discusses the remaining vulnerabilities of the system along with suggested solutions, some of which are proposed by the authors, and ends by discussing the quantum computing threat and efforts toward it. Finally, section 7 introduces suggestions for further future research and section 8 concludes the paper. Extended details on the suggested usage of Verkle Tree are provided in Appendix A.

2 Recent Opposition Activities with Technical Merit

A technical incident that gained some publicity in 2023 elections [8] was done by the same computer scientist observer[4] in [4]; he ***voted using his own Python code*** [9,10]. This gives an alarm that the voting application, which voters should download to deliver their vote, is not authenticated by the system; the *OSCE* report [2, page 8] believes the incident "*could present a cyber security risk*". The report also mentions *some wrong district votes* that were corrected except one vote, but this was magnified by the opposition [11]. It is the authors' impression from all read material that most submitted complaints get rejected based on passing a *3-days from election* deadline without objective investigation, then the vulnerability gets handled and fixed silently in the following election.

A recent complaint about ***the decryption of invalid votes*** after the current 2024 European elections was also rejected objectively in [12]. Among the three listed

[3] We clarify that although [7] is written in Canada, it is about Estonians being able to vote remotely while in Canada and the risks involved. We also note that there was a question in [8] about the fears of Russian interference or taking advantage of the IVXV vulnerabilities.

[4] The word "***observer***" is a term used by IVXV to acquire certain access rights during the election (as opposed to "auditors" as we will detail shortly). Also, the term "***Computer Scientist***" taken from [5] is extremely rejected by IVXV representatives who describe Mart Poder as "***A hobby hacker activist***"; the OSCE report referred to him as "***someone with sufficient programming skills***"

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 [13]. According to [14, Conclusion-page 60], generating proofs of correct decryption of invalid votes was remediated in code by the thesis writer to auditors only in 2024, but *the file containing the decryption of invalid votes is only accessible to auditors* [14, page 22]; this will be further discussed in section 5.1.

The recent European parliament elections were accompanied by some newer actions from the i-voting opposing community [5,12,15,16[5],17]. The same observer mentioned above has developed some kind of ***shadow e-voting*** site called *virtual threshold survey* [15] encouraging citizens to vote again on it as a check (although no evidence of considerable participation ratio).

An earlier complaint about ***the election desktop*** is also alarming and worth mentioning; a first complaint granted an observer (on 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 [18]. The observer took the photo shown in Fig.1 when the disk inspection took place (28/11/2023); it can be concluded from [17] that he pursued the matter further to the supreme court where they responded that "*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*"[6]. Later, on Aug 2024, a commentator from IVXV team stated (in an earlier reviewers' report) that they admit the risk and that it had been taken care of.

Fig.1. image taken from [17]; according to the observer, this is the computer used in key creation which was supposed to have an authentic Windows10 operating system, but the operator didn't notice that DigiDoc4, Notepad++ and RamDisk tools are also installed on it.

Finally, although not an opposition activity, another scientific report from ***the Cyber Security Committee of the Academy of Sciences*** has been handed to the election

[5] People from the system clarified that although AGO Samoson is a respectful researcher in Tallin University of Technology (TUT) School of Information Technologies, TalTech, he is specialized in NMR and materials science (https://www.researchgate.net/profile/Ago-Samoson). They also clarified that "*Cybernetics* is NOT the same institution as *Cybernetica* (even though it shares some common history, but this ended more than 25 years ago) and the researcher Ago Samoson is in no way affiliated with Cybernetica, nor IVXV development", where *Cybernetics* split in from *Cybernetica* (the company behind the current Estonian internet voting system since 2014 partnering with *Smartmatic*).

[6] The official progress of events is in (https://www.riigikohus.ee/et/lahendid/?asjaNr=5-23-40/2)

organizers [19]. The complete report is not published, but the minutes of the last committee meeting [20] on 3rd of June 2024 clarifies they *have identified **6 threats** whose risk class is higher than small*[7].

3 A Brief on System Evolution

As mentioned earlier, digitization has been in Estonia for more than 20 years, even before 2001, and has 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. More details on digital system architecture and components like *Xroad*, *KSI* private blockchain is out of the scope of this paper and can be found in [21]; however, we find the e-ID key generation relevant since it is used in internet voting from its beginning in 2005 up till now. 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 on 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. Such "rare" incidences where citizens were asked to re-install the Java Applet on the cards at PPA (the issuing authority) stations (otherwise the card transactions will be suspended after a certain time limit), became more frequent with time; hence providing more data & information for researchers to analyze. Then, it was proven that the ID card manufacturing company, ***Gemalto***, generated the RSA keys outside the chip (could be to fasten the process) which violates the agreement rules and gives a chance for the key pairs to be copied and repeated. A lot of interesting details on how the analysis was done can be found in the presentation [22] and the paper itself [23]; more faulty keys issues[8] can be found in the PhD of the same researcher Arnis Parsovs [24], and in [25]. Also, other RSA vulnerabilities were discovered in [26].

According to [1], this was a global crisis for the company which was sued in many other countries around the world; Spain and Slovakia [27] replaced all the physical cards while Estonia fixed them remotely. Then, they changed the company to ***IDEMIA*** [28] and, as recommended by [22], moved to threshold cryptography and homomorphic encryption; the Estonian i-voting system IVXV also uses ***384-bit Elliptic Curve*** Cryptography ECC with El-Gamal Encryption, but the list of

[7] A committee member commented (in a non-publicly available statement) on 14/8/2024 that all the 6 threats are of risk class medium (11-13).

[8] Example errors include codes printed too dark which made them readable using torch, without opening envelope (happened twice 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.

authorized votes is still signed using 2048-bit RSA key. Note that, this means Estonian digital IDs will also have to migrate into post-quantum plans soon.

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.
According to all available references, the main design theme of a *double envelope protocol* sending voter signed (first encrypted by the election public key) ballot to the vote collector has been there since 2005. Then, based on [sec.1 of 29,30,31], we mark 2 milestone step transitions:
-In 2011, a student named *Paavo Pihelgas*[9] demonstrated a proof-of concept ballot-manipulating software that relied on the absence of an acknowledgement from the vote collector to the voter that his/her vote was received. Hence, *the ability for voters to verify their votes* was first introduced in 2013 in [32]. However, several flaws were discovered in 2014-2016, [33], that could maliciously alter the vote or the QR code; until *Cybernetica* partnered with *Smartmatic* to produce the QR verification code in its current form, [34], in IVXV.
-Then, with the appearance of other comparable e-voting systems (ex. *FLEP* in France & *SwissPost* in Switzerland), rich material was available for cryptographic research and lessons were learned. Hence, since 2017 the Estonian i-voting, IVXV, added a *vote-registration* service to guarantee no vote dropping, a *shuffling re-encryption mix-net* for vote privacy, and a *Schnor based* NIZKPs non-interactive *zero-knowledge proofs of correct decryption* as 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 [35]. Then we detail an important cryptographic attack along with its fix (done before the 2023 elections).

4.1 Brief Factsheet

The developing companies are ***Cybernetica-Smartmatic*** [36]; the voting device must be a desktop PC (mobile voting is still postponed at least to 2025 [37]); voting can be done using *mobile-ID*, *Smart-ID*, or any digital identity integrated in the *web-eID*[10];

[9] According to [30] the student filed a complaint to the Estonian Supreme court requesting to nullify internet votes in 2011 elections, but his complaint was dismissed for passing the 3 days limit (https://www.riigikohus.ee/en/constitutional-judgment-3-4-1-4-11)

[10] The newer IVXV version used in EP-2024 included extra ***web-eID assistance service***, ***Smart-ID assistance service***, and more other processes to scale horizontally enabling the usage of different digital identities (see section 2 of the architecture file and secs 8.5-8.6 of the protocols file in [39]). The web-eID solution (https://www.id.ee/en/article/web-eid/, https://github.com/web-eid/web-eid-system-architecture-doc) enables the use of different

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; ***Mixnets*** are used to scramble votes before decryption to preserve ballot secrecy. The *election secret key* is divided into parts issued to the members of *the Election Commission of the Republic*, such that decryption requires *5 out of 9 parts*. Finally, there is ***an auditor application*** (could be run by anyone) that verifies the cryptographic 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.2, which could be summarized as follows

1. The voter installs the voting application[11] , sometimes abbreviated as **VA**, on his/her PC.
2. After submitting the digital identity ID, the voting application checks the eligibility of the voter to vote through *the registration service*, **RS**, and if eligible displays the candidate choices for that voter (according to district).
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*[12] 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 (sometimes abbreviated as **VC**).
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* to be mixed and shuffled by mix-nets[13], then decrypted at the counting phase; after voting is closed and before sending to mix-nets, the ballot processor performs some integrity checks, removes multiple votes and votes over ridden by poll station voting.

---

digital identities available in Estonia as a part of applying the European Union web-eID project for all public key cryptography digital identities across Europe.

[11] Sometimes called Voter application, but we prefer to make it distinguishable from the voter.

[12] Before, the timestamp certificate was used to distinguish the last vote of each voter and also for checking the possible verify duration of 30 mins; in the 2024 version *the certificate* sent by the Registration to the Collector *is* a ***signed CONFIRMATION*** (by RS) that contains the original request (***ORDER***) sent (and signed) by the VC, along with the *timestamp*.

[13] IVXV uses *Douglas Wikström's Verificatum* (https://www.verificatum.org/); the package itself provides a verification application, and so does IVXV (and several other projects [29])

5. The vote collector sends a verifying ***QR code***[14] to the voter for optional vote verifying (through verification application) using a second smart device.

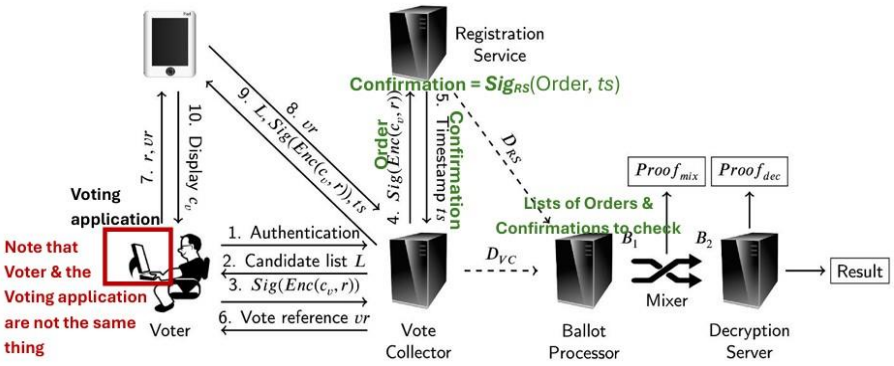

Fig.2. a diagram describing the architecture & the steps of the Estonian voting system, adopted from [1] with some colored remarks in red, while updates in IVXV 2024 version are in green

4.3 Cryptographic Details of Last Fixed Attack

We find it significant, also gives a closer look into the used cryptographic primitives, to explain the exploit introduced in [30].
Let the election public key be "**y**" with corresponding secret key "$\mathbf{S_k}$", and "**g**" be the generator for El-Gamal encryption; hence, the equation « $\mathbf{y= g^{Sk}}$ » holds.
-To encrypt a vote "**v**" the voting application generates a random number "**r**", so that the encrypted vote is $\mathbf{(C_1, C_2)= (g^r, y^r v)}$
-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 the 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 received $\mathbf{C_2}$ 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 fraud if the voting application sent a wrong "***r***" value to the vote collector, ***r'*** such that $\boldsymbol{y^r v = y^{r'} v'}$ to deceive the vote collector into recording ***v'*** as the voter's intended vote.

Long story short, the authors found *three possible manipulations* all with *a simple fix:* making the vote collector send the whole encrypted pair (C_1,C_2) to the verification application which should also *verify that* $\boldsymbol{C_1= g^r}$ as was finally done [38, lines 77-83 & 141-146 in code and the exception is thrown at line 60] on 23rd Feb 2023 just

[14] According to [9], there were a revealing incident of *the president vote* through his QR code: he voted online in front of cameras, showing his QR code, to encourage citizens; someone took a snapshot of the QR code and revealed his vote. The incident was mentioned in the context of doubting privacy and hence protection from coercion and/or vote buying.

before March 2023 elections. The authors alarmed that it is concerning [30, sec 3.6] that such a straightforward vulnerability wasn't noticed earlier, and then criticized the quality of IVXV in general [30, sec.4].

5 Enhancements Against Vulnerabilities

In addition to extending IVXV to support voting with more kinds of digital identities like web-id and mobile-id, [39], the European parliament election version IVXV 1.9.10-EP2024 included many other improvements. This section details three of them which targeted discovered and/or criticized exploits.

5.1 Decryption of Invalid Votes

Another improvement was added to IVXV on 30th May 2024 [39], just before the European Parliamentary elections on 3rd of June; invalid votes are 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 verify those proofs as we discussed the earlier complaint.

Why not reveal invalid votes?

As mentioned in section 2, there were a lot of debating and complaints about not allowing observers to view the decryption of invalid votes; however, the reasons stated by the state election service in the supreme court decision [12] do not clarify the risks as in [14].

-Reasons 1&2 in [12] talk about the technical infeasibility of decrypting invalid votes after the election and how this needs parts of the secret election key (issued only to members of the election commission); on the other hand [14] explains how the IVXV version used in 2024 already decrypts invalid votes in a separate file, and this can be traced in the opensource code [39/key]. In general, election data gets destroyed a month after the election.

-The 3rd reason in [12] of "*not knowing in advance what the invalid ballot contains and it may be an attack*" is rationalized better in [14] as the possible reveal of some information about the voter of the invalid vote, or more severe the threat of ***encoding attacks*** described in [40, sections 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 not even needed if invalid votes were rejected earlier by the vote collector as [14] suggests. In fact, tracing the *number of* ***invalid votes*** in the official statistical site [4] to be ***exactly 1***[15] in the last three local elections since 2021 makes it look quite suspicious; the doubt includes anyone who can see the votes and should be eliminated completely when range proofs get deployed.

[15] The number became 2 in the European elections.

5.2 Integrity Checks

Another problem that was mentioned in the *OSCE* report [2] is that there's no cryptographic proof for *the deletion of multiple votes or ill-formed vote ballots*; i.e., the authorities are assumed trusted regarding not deleting or adding extra votes at this step. 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 [41] as trusting the vote collector and registration applications to not collude, otherwise it would be possible to drop ballots; the *Ballot Processor* in Fig.2 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 [29], 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 (section 6.4) is a partial protection from this vulnerability was integrated into the audit application of IVXV through a file named ***Integritytool.java***, [42], published on 30/5/2024 just before the European parliament election; however, the details of the solution were only published academically in [43] 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 [43/section 3.A] that performs some processing on this data (step 4 in section 4.2) to then input the list anonymized votes to the mix-nets; Fig.3.

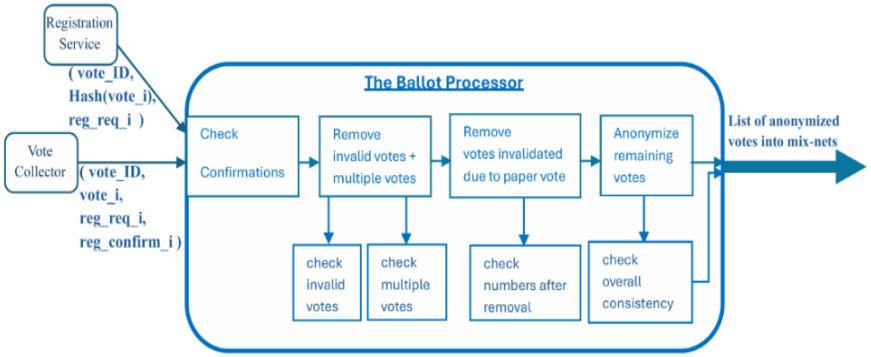

Fig.3. The (modified) processing stage after adding the *integritytool.java* file

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 [36/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 [38]. Hence, what was ***mitigated*** by [40,43] is ***the risk of a colluding Ballot Processor***.

5.3 Session ID check

A possible attack by a malicious voting application that could deceive even verifying voters was discovered by Olivier Pereira in [31]; a malicious 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 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

-The author suggested a few mitigations, we have no clue that any of them was adopted, except storing a *voteID* field with each vote.

-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 [44], especially if their device suffered a system crash while voting.

-Although it is complicated to design, since it requires searching the final anonymized list of votes, the QR code could contain a flag on whether this is the last vote or not[16].

-Another simple safeguard from this specific problem, [45], 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 [31] and a malicious voting application would have enough time to submit without voter's knowledge.

IVXV Adopted Solution

Although not publicly advertised, a solution to this attack was added on 30th May 2024. The code was updated to simply check the session-ID is still the same before generating the verifying QR; the code documentation [46, lines 22&103] highlights that this "***prevents reusing session ID until it is deleted from a database or expired***".

Tracing a little deeper, [39/protocols PDF/sec. 8], the ***PKIX*** *(Public Key Infrastructure X.509)*[17] timestamp protocol is used by the registration service to

[16] The timestamp alone is not enough to help the voter detect a problem; in case of a faked system crash, the voter may not notice the difference and say "OK, maybe the system records the time when I started the voting attempt, not when it has ended". The number of this vote wouldn't help also, as it won't tell if there were later votes.

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

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. However, according to [47], the two methods *SessionStatus.Read* and *verifyStatusReadResp* can detect a fake restart if the server expires and revokes the same Session-ID afterwards. The current code ***relies on timestamps expiration***, so a malicious application reusing an unexpired Session-ID could slip through unless tighter server-side tracking is in place.

6 Remaining Vulnerabilities/Issues

This section starts with a preface that summarizes the status quo of the Estonian electronic voting systems, then follows with some remaining vulnerabilities and proposed mitigations (whether by the authors or in previous literature). As we expect IVXV is approaching the end of an era and will have to face the quantum computing threat very soon; we will discuss the post-quantum version of our suggestions as we go, then seal with possible post-quantum system upgrades.

6.1 The Status Quo of IVXV

Many academic research papers, in addition to pointing out some attacks [30,31,40], have introduced a holistic criticism to IVXV; [30] analyzed *IVXV public information* as *satisfying* only *1* (minimal restriction on disclosure of vulnerabilities) *out of 9 quality metrics*[18] and warned from the possible existence of hidden vulnerabilities; [40] demonstrated (through the analysis of possible privacy attacks) that *IVXV is vulnerable to attacks against vote privacy in those threat scenarios that were considered for it originally*; [41, sec. 5.1] discussed the different trust assumptions of IVXV including software components and key holders, in this context [40] also discussed that Vote Collector is trusted on the privacy of encrypted votes.

Although those papers were all written in 2023 analyzing the version used in June 2023 election or earlier, and although we did trace many improvements in the current 2024 version IVXV 1.9.10-EP2024[19] meaning we could have missed some, we know the newest version is the one analyzed in the most recent report [19,20] which identified *6 threats with risk level higher than small*. The report was done in collaboration with the election authorities (i.e. not biased against IVXV); also, the *OSCE* 2023 report [2] pointed out to some issues only some of which were resolved.

Although the IVXV team sticks to the fact that there was no proved error in the election results, we point out to the low *QR verification ratio of 5.5-9.9%* as stated in

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

[19] GitHub history shows 897 changed files with 34,059 additions & 10,830 deletions. Translating [39] from the Estonian language, a lot of work was done in integrating different kinds of digital identities and in coordinating with XRoad service (X-tree). A whole section is dedicated to the Registration Service, [39/protocols PDF/sec.6]; the interaction between online (RIA), offline (RVT) and other IVXV services.

the official i-voting statistics site [4]; this does not really prove beyond reasonable doubt that no pairs were manipulated.

Also, as a general attitude, IVXV does not advertise its progress to researchers; the improvements in section 5 were discovered through repeatedly scanning the academic literature and/or tracing the code; the authors of [30] were not informed of the fix in section 4.3 till their paper was published; when the authors of this paper contacted them in Feb 2024, [45], they did not mention that they are already working on the mitigations deployed on 30th May 2024. The official English site, [48], still have the new version files only in the Estonian language after more than 11 months.

6.2 The Voting Application

Even if the IVXV solution discussed in section 5.3, [46], completely protects from the Olivier Pereira attack, [31], it is not a general solution to the voting application vulnerability. The authors of [30] commented on the voting application being the only unrevealed part of IVXV code as an open source, while [41] identify this fact as a trust assumption. What is more of a threat is the incident of overriding it in 2023 election, [8,9], which proves that it is not even authenticated; accordingly, the *OSCE* report [2] notified about the risk of not authenticating the voting application. A clear obvious vulnerability here comes from *the possibility of downloading a* ***malicious voting application***; this leaves it as *an open challenge for adversaries* to design the most possible malicious code they can come up with. Having a ~ 90-95% probability that the voter will not verify the vote, an adversary could use social engineering to target those who are not likely to verify which makes the risk level more severe.

Another possible risk is *for* ***vote buyers/coercers*** *to do what the authorities haven't done*; i.e., develop a fixed candidate voting application and authenticate its usage through *execution attestation* on the voter PC[20] before transferring the money. This *DarkDAO*s idea was discovered by [49] in 2018 as a possible threat to decentralized voting in DAOs using governance tokens, but it could happen here too; the authors published a follow-up in 2023 [50] with a GitHub code[21].

A general solution to all the above is ***to authenticate the official voting application***

-A simple moderate safeguard is to publish its *file digest* (hash SHA256 for its code for example) and encourage voters to run a check before using it; the *Electrum* Bitcoin wallet already gives this option when downloading [51], so it could be easily implemented. Although this is still very much voter dependent, one can count on users performing system recommended pre-checks with higher ratio than post-checks

[20] Remote execution attests were originally discussed on Intel SGX (available on many new PCs in the market: https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions-processors.html), also supported by other companies like Apple. So, there's a considerable probability that the voter PC could have it.

[21] The same authors next participated in (https://medium.com/initc3org/complete-knowledge-eecdda172a81, https://eprint.iacr.org/2023/044,) which suggests that a TEE only provides execution attests after submitting ***proof of Complete Knowledge (CK***) of the user device.

(like the ≤10% ratio of QR codes). As a second safeguard for QR-code checkers, the verification application could also display an extra message with the vote saying that "*You voted using (the official/a different) voting application*". This fulfils the system philosophy[22] of giving suspicious voters the freedom to code and use their own voting application (or maybe a one written by a political party they trust more).
-Another solution we believe is more intact (since it is not optional) is to assign *a signature & authentication key pair* to the voting application like the rest of applications in the system. Then the election authority can allow only usage of *a pre-registered private voting applications* with a *stored public key* at the voting server; this way, the election authority can also *scan* the private voting applications for any malicious or vote buying code before granting usage rights. However, another issue remains in this solution in ***how to inform a non-verifying voter that the vote was rejected because he/she has installed a malicious voting application***.

We think, [45], *the vote collector application could deduce the IP address* of the voter machine from the first contact with the voting application, then it can *send a* direct warning message; something that pops up on the voter screen "Be careful, this is not the official voting application". The use of *a feedback channel* was also one of the suggested mitigations in Olivier Pereira paper [31] and a clear message about the application can mitigate his fears of deliberately delaying; it as long as it is received before voting is closed, the voter can vote again either through a more secure device or in a poll station. Although the authors of this paper are not very familiar with the technicalities of the down network layers involved in implementing this informing mechanism, it is feasible to implement[23]; IVXV already acquires knowledge of the IP address to calculate ratio of abroad voters as stated in [4].
Finally, all those solutions could be easily made post-quantum by using post-quantum hash based digital signatures, like *Stateless Hash* (*SLH)-DSA* approved NIST standard (will be detailed in section 6.6), since its relatively large size of 7k Byte is only transmitted once with the download and won't be noticeable by users [52/min64].

6.3 Range Proofs

To eliminate any complaints (like [12]) and/or risks (ex. [40]) accompanying invalid votes, it is much safer to prevent them from reaching the Ballot processor at all. The thesis in [14] 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

[22] Although the response in [45] does not give this impression, some of IVXV team respond that being able to vote with your application contributes as a measure to the transparency of the system in Estonia; in any case, the solutions suggested here support this feature.

[23] The last few lines in (https://stackoverflow.com/questions/35301392/how-to-access-a-remote-desktop-from-a-virtual-machine-set-on-a-server) show that similar things have been done, and (https://serverfault.com/questions/229216/application-which-can-pop-up-like-gtalk-when-some-one-accesses-my-server). Also, any other service in the Estonian e-government that links electronic IDs to cellular numbers or emails can receive just the voter ID to send a fixed message "*Beware you are not using the official voting application*".

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*** **&** ***Pederson 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 original paper for Microsoft *ElectionGuard*, [53], also assumed the existence of NIZK (Non-Interactive Zero Knowledge) Range Proof for ballot correctness in their system; the implementation of *ElectionGuard* v.2 (open source) used ***disjunctive Chaum-Pederson*** Range Proof[24].
-We could add that whatever the underlying cryptography is, ZKPs that are based on the discrete logarithm problem (like *bullet proofs* and *Chaum-Pederson*) enjoy faster prover time than general purpose SNARKs. This is a desirable quality for ballot validity proof since the voting application is the prover (the vote collector is the verifier); the prover code must be executed online during the voting session and cannot be batched or pipelined even if the used ZKP allows it.
-However, a recent paper, [54], (a follow up to *Kryvos* referred to by [14] in the subject) introduced benchmarks for an efficient implementation of *Groth16* over El-Gamal based voting systems. The authors used a 254-bit common elliptic curve BN254 in their implementation, while IVXV uses 384-bit curve; note that the circuit specific setup needed in Groth16 could be considered a drawback.
-Another 2024 paper, [55], introduces *Polymath* proving it can be more efficient than Groth16 and relates it to KZG commitments[25]. Here, the authors did compare KZG and Groth16 over a 381-bit curve (BLS12-381) which is closer to the one used in IVXV; their work also included mathematical proofs for batching reductions.
Finally, all of this will have to be rethought-of seeking post-quantum validity proofs. There are many post-quantum NIZKs and SNARKs, but the choice must be compatible with the new post-quantum encryption that will be used.

6.4 Insiders' Risk (The Number of Ballots)

We mentioned in section 5.2 that a solution was already implemented to overcome insiders' risks at the processing stage. However, the integrity of the input file still depends on the vote collector and registration application not to collude, which is also an insiders' risk; for example, the Count values are not cryptographically proved. For this vulnerability, we suggest two alternatives; the first depends on checking the consistency of different data sources available in the Estonian government.
Overall Checks

[24] https://github.com/microsoft/electionguard-rust/blob/main/TODO.txt#L1373
[25] A condensed summary and a comparison between possible SNARK choices can be found in the first 25 mins of (https://youtu.be/A3edAQDPnDY)

This solution aligns with IVXV solution discussed in section 5.2, since the ***PKIX*** timestamp protocol is used to both record the time of casting the vote and to register the electronic vote in an external independent service. So, instead of trusting the initial *Count* value coming from the vote collector and the registration service, extends to check it with other data records available in the Estonian e-government.

-A possible general *double check* for the whole list of votes is to compare with the transaction records of the Estonian Information System. In fact, there is an existing *myID* service, [44], that provides what could be viewed as an individual verifiability double check for voters using the *eID* digital identity[26]. We suggest that IVXV performs similar universal verifiability checks on all votes; i.e., checking that *the total number of transactions to IVXV services equals the total number of existing ballots*. Recalling Eq(1), we add the check:

Verify:
Count (original votes file) =
Count_Transactions (source=all, destination=IVXV, time=election_interval)

-If there exist aggregating queries on the information system, the same above check should be repeated for checking the integrity (not just the count) of all votes. The verification process could be a simple hash cascade, a sophisticated ZKP, or even a comparison between sorted versions of the common fields between the two lists:

Verify:
(original votes file) =
Transactions (source=all, destination=IVXV, time=election_interval)

-If aggregating queries were not possible, the first check could be accompanied by some kind of ***Risk Limiting Audits (RLA)***s, where only a sample of random votes could be selected to check manually. RLAs are usually used in e-voting systems that deploy dual paper ballots, [56]; here, transactions stored in the Estonian information system could play the role of paper ballots to compare with IVXV data.

```
for all i ∈ sample
{   n= Count (original votes file, vote_ID=i);
    Verify:
    n = Count_Transactions (source=i, destination=IVXV, time=election_interval);
    for all j = 1 to n            //in time order
    Verify:
  Original votes file(vote_ID=i, order=j)=Transaction((source=i, destination=IVXV);
}
```

6.5 The proposed Verkle Tree

A more robust protection from insiders' risk is to construct an online SNARK with each vote; i.e., cryptographically commit to the vote spontaneously. This choice can be also favorable if the Estonian government does not prefer the interaction between the e-voting system and other e-government entities and yet cares to cryptographically

[26] Though in a different context, [9] shows and demonstrated through X conversation (https://x.com/trtram/status/1763936733027049606) how a sophisticated user can do that.

prove the counter values (for each voter and for the whole number of votes) in addition to the data integrity.

We suggest ***aggregating vote hashes in an Authenticated Data Structure ADS*** [57, sec.2.1 def.3] which we can simply describe as data structures that can provide succinct (short) cryptographic proof (sometimes called witness) of each element stored in them and ***that can cryptographically prove the number of values stored in it.*** In fact, [40] has earlier suggested as a protection from privacy attacks that each voter computes a *NIZKP of knowledge* of his/her encrypted vote, and the newer version of IVXV partially responded. Currently, the Registration Service sends *hash(vote)* to the Ballot Processor (Fig.s 3&5). Tracing the code in [42, 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 the tree (votes) is not cryptographically verified.

In this paper, we suggest the use of ***Verkle Trees*** [58], which is a vector data structure that authenticate its elements based on KZG polynomial commitments (as the polynomial coefficients) because they have the shortest verifying complexity (constant order), more important because they provide a cryptographic proof of the number of elements stored in them (as opposed to Merkle Trees) which is the number of votes in our case; also, the data structure could be virtual, only the needed proves have to be kept in storage after runtime. The longer prover time for general purpose SNARKs, mentioned in section 6.3, is not as problematic here, since proofs can be constructed in the background and /or batched using the homomorphic property. Hence, we propose to aggregate all votes in a Verkle Tree (VT); every used Verkle Tree will add a line or 2 to the code {***VT=VT+H[i] * committed_vote***; *i*++;}, where the vector H is calculated during setup (Appendix A.1).

-If this was used in conjunction with the solution in section 5.2, then we are done here; the Ballot Processor can verify the VT value in its first block (Fig.5), and RLA samples can be verified similarly. If the VT value were published instantaneously when the voting is closed, and the original list of all votes hashes is also available, even independent anonymous verifiers can verify it too. It is also important to make the code publicly available to guarantee the integrity of the VTs construction steps.

-If this solution is to be used to further validate the removal of multiple votes, then it must construct another dynamic final votes VT_2 in parallel; votes from the same voter could be moved to aggregate in second level VTs if needed or accumulated all into a third VT_3 otherwise. There could be a variety of ways to implement this idea depending on the needed checks; see Appendix A and Fig.6 for the details, where only step1 is needed in conjunction with [42].

Discussion

Authenticated Data Structures (ADSs) are proposed here as a mean to lose trust assumptions of voting modules by publishing SNARK values that were dynamically calculated using a publicly available code. In this paper, we chose Verkle Tree commitments as opposed to Merkle Trees or STARKs based on the following logic.

-Verkle Trees provide a constant order complexity *SNARK* (per node or per batch that could be all the data) using KZG vector polynomial commitments. Although Verkle Trees require trusted setup procedure to generate a crs (*common reference string)*, but

we do not consider this a problem since it is a universal setup (not circuit specific as in Groth16) which aligns with IVXV setup & key generation phase. This arrangement also allows the use of *Risk Limiting Audits (RLAs*) to verify the details of a selected sample of ballots.

-Traditional Merkle Trees can support checking RLA samples too (the logarithmic complexity is not an issue here since the checks are done offline). However, Merkle trees do not verify the total and subtotals numbers which is crucial in our case.

-Although the ***STARKs*** (*Scalable Transparent ARguments of Knowledge*) option introduced in [57] is based on FRI commitments and hence provides post-quantum security guarantee, and also does not require a trusted setup phase, we find those advantages helpless in our case; IVXV involves a key setup phase anyway, and deploys EL-Gamal encryption which is not post-quantum to begin with. In their paper, the authors showed a projected performance analysis for applying their approach to EL-Gamal based e-voting systems, but it is not a performance issue; if quantum computing is feasible, an adversary can discover the private keys and produce correct values that will pass the verifier checks. Hence, we believe STARKs can only be a better solution if they were a suitable match with a post-quantum update of the underlying encryption.

Finally, there is an interesting intuition from [57] that can be useful in archiving IVXV results. Their suggestion to *make the verifier check the generated proof instead of the original data* can be used when election data gets destroyed (after a month); i.e., the VTs generated in Appendix A could be kept forever as permanent proof.

6.6 Post Quantum Cryptography (PQC)

Quantum computing depends on physics and quantum theory to perform computations much faster than current existing hardware that it could solve all hard problems we currently consider infeasible; specifically, *Shor's Algorithm* [52,59] can solve the Discrete Logarithm and all Elliptic Curve problems, and hence break all cryptographic functions based on their infeasibility. Majority of experts did not believe it is a probable threat [59/Fig.1]; only cryptography researchers discussed it [57,60,61] and started a long and complicated road to post quantum alternatives.

Main Solution Strategies

The *ENISA* (*The European Union Agency for Cybersecurity*) 2021 report, [62], was dedicated to post quantum solutions with a focus on the 3 finalists of NIST round at that time. Suggested solutions [61,62] are based on ***multivariate*** equations over a finite field (ex. UOV and Rainbow), others are based on ***error correcting codes*** (ex. McEliece), but most are based on ***Lattice*** cryptography (ex. Kyber) and/or ***Hash functions***[27] (ex. *SLH-DSA*). ***Lattices***, [63], can be viewed as *n*-dimensional matrices (represent discrete points in an n-dimensional space); this can provide problems that remain hard to compute even with quantum computers like *Shortest Integer Solution*

[27] The 2016 NIST report, [61], stated that Symmetric key encryption can tolerate by increasing key size, and SH-2, SH-3 hash functions by increasing the hash length. In any case, hashes are treated as oracles (ROM) in most designs; hence, can be easily replaced by PQ ones.

(SIS) and *Learning With Errors (LWE)*. In 2024, [64], the NIST announced 3 PQC standards; a hash based one for digital signatures (*SLH-DSA*), and two Module Lattice-based *ML-DSA* for digital signatures and *ML-KEM* for Key Encapsulation. However, security analysis of *CRYSTALS-Kyber*, the one *ML-KEM* is based on, including vulnerabilities and a *side channel* attack[28] can be found in [65,66].

Threat is Approaching

-Other than research, Quantum computers remained just a theoretical threat that is infeasible to implement; even when believed otherwise, none of the interviewed experts in [67] (May 2024) expects upgrading embedded infrastructure devices to take less than ten years[29]. At the end of last year, things got accelerated rapidly in a way that raised awareness; on Dec 2024, Google & Microsoft unveiled their first chips [68] (although exaggerated by the media [52]); on April 2025 a quantum computing group [69] offered a 1 BTC (i.e. ~100,000$ anticipated cost) reward competition to break a Bitcoin key using quantum computing before April 2026. European financial corporations announced $1bn investments plans on *PQC* in 2025 [70], as opposed to $162.8m value in 2024 [71]; also, NIST latest report anticipates $7bn worldwide [65].

-In general, the retaliation game for political elections is more severe than that of financial systems; some countries are willing to pay much more to break a political election than a thief wanting to steal money. In addition, there is the *Harvest Now Decrypt Later (HNDL)* threat of storing encrypted data to decrypt them later when QC is possible [52]. Hence, all countries deploying e-voting/e-government systems should be more motivated, without uncarefully studied rush, to provide post-quantum solutions; in fact, e-governments/e-IDs would be a probable main target of a quantum adversary than e-voting systems as stated by most interviewed experts in [67].

Estonian Quantum Efforts

For our specific case of Estonia, the authors of [67] included 5 out of 24 experts from Estonia in their interviews, and covered steps taken by the Estonian government [67/ sec. 2.2.2] in different quantum related aspects; a European cooperation to deploy a quantum communication infrastructure in 2020 [72] which led to the NordIQuEst (*Nordic-Estonian Quantum Computing e-Infrastructure Quest*) in 2022 [73], a collaboration between *Cybernetica* (IVXV company) and University of Tartu to create post quantum solutions has started in 2021 [74] including jointly supervised doctoral thesis from which we encountered a paper (2024) on PQC migration obstacles in Estonia [59], and 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 [75]. Then, the European Commission published on 11 April 2024, [76], a coordinated PQC implementation roadmap for all member countries. The announcement contained only one guiding statement; the implementation should be "*via hybrid schemes that may combine Post-Quantum*

[28] Side-channel attacks, [65], were first identified by Paul Kocher in 1996 as *Timing attacks* on implementations. They learn information from data disclosed while a cryptographic device is in use like electromagnetic radiation, sound waves, power use, or execution time.

[29] An Estonian research paper, [59], puts the first obstacle to post quantum transition as: "*The urgency of starting the post-quantum migration is not well understood by decision-makers and those outside the cryptographic community*", while [52] warns from unstudied rushing.

Cryptography with existing cryptographic approaches or with Quantum Key Distribution (QKD)". This was different from ENISA 2021 report [62/page 35] which excluded solutions using *QKD* because its setup relies on pre-established authenticated communications channels; [67,77] also reported different experts' opinions on this point. In fact, [59] considered "*EuroQCI focused attention on QKD technology"* as an obstacle to post quantum transition since they believe its functionality is limited compared to PCQ.

Estonian I-Voting Related Efforts

-Digital IDs are a crucial element in i-voting; the *ENISA* July 2024 report [78] on the unified European digital identity project *eIDAS*, though, did not specify the PQC transition details. Then on 19 Feb 2025, [79], *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 [59/sec. 2.E], they considered the complexity of managing multiple layers a disadvantage in most existing Hybrid systems. For i-voting, [59/sec. 4.E], this would be a problem with the needed multiple layers to shuffle votes in *mix-nets*; we think lattice-based approaches maybe advantageous here, [77], as they depend on matrix operations. Brief survey on post quantum i-voting research attempts, and some post quantum replacements for El-Gamal encryption is in [80].

7 Possible Research Directions

In this section we go through possible research threads that could come up from the gathered material, even if not necessarily pursued by the authors of this paper.

The work in [54] could be a starting point for researchers concerned with hardware acceleration of Zero Knowledge Proofs that goes in depth into circuit specific details, while [55] would be useful for researchers interested in theoretical cryptographic proofs. Also, research can be further pursued to conduct comparative analysis between possible zero knowledge proofs concerning implementation details along proof batching reductions over the exact 384-bit elliptic curve used in IVXV.

However, one may think the future will promote more research on efficient quantum secure solutions and less encouragement to elliptic curve research. The research space is challenging, [61], on the efficiency-security tradeoff; demanding for new innovations [63], for cryptanalysis [65,66], and formal verification [77] of the new PQ protocols (how to prove correctness and security if it is not verifiable on existing hardware [52]). In addition, AI-based attacks could be another rich research area; side channel attacks, [66], "*how information can be leaked from the implementation of a cryptographic component*" and maybe from quantum devices too.

Finally, with the intuition of [49], it could be an interesting research to tackle the broader question of *to what limit can the information provided by general purpose activity logs of digital identities* (in Estonia or any country that uses digital identities in online voting) *help vote buyers and/or coercers* in catching voters who try to deceive them, and whether a blockchain based e-government is an advantage or disadvantage in that direction. We have already discussed in section 6.4 how [44] and

similar services can impose this trade off; to what limit could adversaries expand their toolbox using such services to defeat, for example, the *CK* (*Complete Knowledge*) suggested metric whether for vote buying/coercion or any other malicious activity.

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. Then we explained the current system architecture and surveyed available material from the academic literature and different other available resources to cover reported attacks and/or vulnerabilities and how they were fixed by the system. Last but not least, we discussed remaining vulnerabilities; mainly, authenticating the voting application and cryptographically proving the list of votes against any possible insiders' manipulation. We suggested some possible solutions that give voters the freedom to deliberately use a different voting application they trust more; the first is allowing the voter to check the application fingerprint prior to voting; the second is allowing only a list of pre-registered applications (through digital signatures) and informing the voter with the result. Then, we proposed alternatives for more robust guarantees that the list of votes was not manipulated inside the system; one is to perform consistency checks and RLAs between ballots in IVXV and with digital transactions interacting with IVXV through the Estonian Information System. The second is aggregating votes (online during the voting process) in an authenticated data structure like Verkle Trees which we detailed as our proposed solution. Finally, we highlighted the quantum computing threat on e-voting systems, and digital identities in general, then introduced a variety of possible research threads that could evolve from all the introduced material.

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. 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.
5. https://gafgaf.infoaed.ee/en/posts/great-divide-in-evoting/; last accessed 14/3/2024.
6. 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.
7. https://electionsnovascotia.ca/PictouWestByElectionEBallot, last accessed 22/4/2024.
8. "*A computer scientist made available the code for e-elections, which the electoral service has so far been fiercely hidin*g", https://digi.geenius.ee/eksklusiiv/arvutiteadlane-tegi-

kattesaadavaks-e-valimiste-koodi-mida-valimisteenistus-on-seni-kiivalt-varjanud/; last accessed 2/1/2024.
9. https://gafgaf.infoaed.ee/en/posts/perils-of-electronic-voting/; last accessed 4/1/2024.
10. https://media.ccc.de/v/37c3-12298-should_e-voting_experience_of_estonia_be_copied#t=965; last accessed 15/1/2024.
11. "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.
12. 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.
13. https://www.valimised.ee/en/internet-voting/observing-auditing-testing; last accessed 5/7/2024.
14. Taaniel Kraavi, "*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
15. "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.
16. 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,
17. "*E-voting system Disk appeared out of nowhere*", https://gafgaf.infoaed.ee/posts/esoteeriline-turvamudel/; last accessed 22/5/2024.
18. Election Commission of the Republic, "*Review of Mart Podra's Complaint*", 23/2/2023, https://www.riigiteataja.ee/akt/328022023004; last accessed 7/7/2024.
19. The report of *the cyber security committee of the Academy of Sciences*, https://x.com/danbogdanov/status/1802998209649762582; last accessed 6/7/2024.
20. Cyber Security Commission minutes of meetings, https://www.akadeemia.ee/akadeemia/noukogud-ja-komisjonid/kuberturvalisuse-komisjon/; last accessed 7/7/2024.
21. https://ec.europa.eu/digital-building-blocks/wikis/pages/viewpage.action?pageId=533365949; last accessed 28/12/2023, https://scoop4c.eu/cases/estonian-internet-voting; last accessed 22/11/2023.
22. Arnis Parsovs, "*Estonian Electronic Identity Card: Security Flaws in Key Management*"; video https://www.usenix.org/conference/usenixsecurity20/presentation/parsovs
23. Arnis Parsovs, "*Estonian Electronic Identity Card: Security Flaws in Key Management*", 29th USENIX Security Symposium, Aug 2020, 978-1-939133-17-5.
24. Arnis Parsovs, "Estonian Electronic Identity Card and its Security Challenges", PhD Thesis, University of Tartu.
25. 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.
26. 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
27. https://e-estonia.com/raulwalter-estonia-digital-identity-giant/; last accessed 20/3/2024
28. https://e-estonia.com/estonia-introduced-a-new-id-card/; last accessed 20/3/2024.

29. 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
30. 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. hal-04216242; https://halscience/hal-04216242
31. Olivier Pereira, https://www.researchgate.net/publication/372570425_Individual_Verifiability_and_Revoting_in_the_Estonian_Internet_Voting_System
32. 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
33. D. Springall et al., "*Security Analysis of the Estonian Internet Voting System*", Nov 2014, 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
34. https://research.cyber.ee/~janwil/publ/ivxv-evoteid.pdf
35. https://valimised.ee/sites/default/files/2023-02/IVXV-protocols.pdf
36. Smartmatic-Cybernetica. IVXV Voting Service. Version 1.8.2-RK2023, https://github.com/valimised/ivxv/tree/master
37. https://news.err.ee/1609194064/mobile-voting-likely-to-arrive-in-estonia-in-2025; last accessed 14/12/2023.
38. 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 20/2/2024.
39. The *key application*, IVXV 1.9.10 EP2024, https://github.com/vvk-ehk/ivxv/tree/master/key; last accessed 8/7/2024, *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
40. Müller, J. (2023). "*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. Lecture Notes in Computer Science, vol 13412. Springer, Cham. https://doi.org/10.1007/978-3-031-32415-4_22
41. Krips, K., Snetkov, N., Vakarjuk, J., Willemson, J. *"Trust Assumptions in Voting Systems"*. In: Katsikas, S., et al. Computer Security. ESORICS 2023 International Workshops. ESORICS 2023. Lecture Notes in Computer Science, vol 14399. Springer, Cham, https://doi.org/10.1007/978-3-031-54129-2_18; full paper available at https://arxiv.org/pdf/2309.10391
42. https://github.com/valimised/ivxv/blob/published/auditor/src/main/java/ee/ivxv/audit/tools/IntegrityTool.java; last accessed 24/2/2025.
43. 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; published 23/12/2024, last accessed 13/4/2025
44. https://myid.skidsolutions.eu/en; last accessed

45. https://github.com/DrShymaa2022/articles_papers/blob/main/Letter_to_Estonia_ivoting.pdf
46. https://github.com/valimised/ivxv/blob/published/voting/internal/sessionstatus/rpc/client.go(#L22,#L103); last accessed 14/4/2025.
47. https://x.com/i/grok?conversation=1902860748558135338; last accessed 14/4/2025.
48. https://www.valimised.ee/en/internet-voting/documents-about-internet-voting; last accessed 17/4/2025.
49. PMPhilip Daian, Tyler Kell, Ian Miers, and Ari Juels; July 2018; https://hackingdistributed.com/2018/07/02/on-chain-vote-buying/
50. James Austgen, Andrés Fábrega, Sarah Allen, Kushal Babel, Mahimna Kelkar, Ari Juels, "*DAO Decentralization: Voting-Bloc Entropy, Bribery, and Dark DAOs*", Nov 2023; https://arxiv.org/abs/2311.03530; https://github.com/DAO-Decentralization/dark-dao/tree/main; last accessed 20/3/2024.
51. https://bitcoinelectrum.com/how-to-verify-your-electrum-download/, https://en.bitcoin.it/wiki/Electrum#:~:text=8%20References-,Verifying%20Electrum%20Binaries,files%20were%20not%20tampered%20with.; last accessed 17/3/2025
52. https://a16zcrypto.com/posts/podcast/quantum-computing-what-when-where-how-facts-vs-fiction/; last accessed 17/5/2025.
53. J. Benaloh, M. Naehrig, O. Pereira, and D. S. Wallach, "*ElectionGuard: a Cryptographic Toolkit to Enable Verifiable Elections*", June, 2024, https://eprint.iacr.org/2024/955, https://www.electionguard.vote/spec/; last accessed 13/7/2024.
54. N. Huber et al, "*ZK-SNARKs for Ballot Validity: A Feasibility Study*", E-Vote-ID 2024, pp 107-123, Oct 2024; https://link.springer.com/chapter/10.1007/978-3-031-72244-8_7
55. H. Limpaa, "*Polymath: Groth16 Is Not The Limit*", CRYPTO 2024, pp 170-206, Jun 2024; https://link.springer.com/chapter/10.1007/978-3-031-68403-6_6
56. https://verifiedvoting.org/audits/whatisrla/, https//www.sos.state.co.us/pubs/elections/RLA/faqs.html; last accessed 1/5/2025.
57. Max Harrison and Thomas Haines, "On the Applicability of STARKs to Counted-as-Collected Verification in Existing Homomorphically E-Voting Systems", Mar 2024; https://fc24.ifca.ai/voting/papers/Voting24_HH_On_the_Applicability_of_STARKs_to_Counted-as-Collected_Verification_in_Exisitng_Homomorphically_E-Voting_Systems.pdf
58. Zero Knowledge Berkely MOOC 2023, lecture 5, "*KZG polynomial commitment scheme"*; https://youtu.be/tAdLHQVWlUY
59. 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
60. Bello A. Buhari, Afolayan A. Obiniyi, Sahalu Junaidu, and Abubakar Roko, "*Elgamal Cryptographic Scheme based on Quantum Key Distribution (QKD)*", Dec 2020, DOI: 10.47310/iarjet.2020.v01i04.008, https://www.researchgate.net/publication/344784508_Elgamal_Cryptographic_Scheme_based_on_Quantum_Key_Distribution_QKD
61. 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
62. https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation; last accessed 27/4/2025.

63. Dana Sairangazhykyzy Amirkhanova, Maksim Iavich, and Orken Mamyrbayev, "*Lattice-Based Post-Quantum Public Key Encryption Scheme Using ElGamal's Principles*", Jul 2024, Cryptography 2024, 8(3), 31; https://doi.org/10.3390/cryptography8030031
64. https://quantumcomputingreport.com/nist-has-finalized-the-first-three-pqc-algorithms-45-more-are-still-in-the-pipeline/#; last accessed 18/5/2025.
65. Iavich, M. and Kuchukhidze, T., "*Investigating CRYSTALS-Kyber Vulnerabilities: Attack Analysis and Mitigation*", Cryptography 2024; https://www.mdpi.com/2410-387X/8/2/15
66. Grünfeld, J. Mathias H., "*Side-Channel Attacks on CRYSTALS Kyber: An Analysis of a Post-Quantum Algorithm and Its Vulnerabilities to Side-channel Attacks*", Master's Thesis, NTNU, Trondheim, Norway, 2023; (cross reference from [60]).
67. A.R. Perez, N. Costa, and T. Finogina, "*An electoral exception? Quantum computing - readiness and internet voting*", 31 May 2024, JeDEM Issue 16 (3): 50-79, 2024, DOI: 10.29379/jedem.v16i3.928
68. Chris Vallance, "*Google unveils 'mind-boggling' quantum computing chip*", https://www.bbc.com/news/articles/c791ng0zvl3o, last accessed 22/4/2025; Google published paper: https://www.nature.com/articles/s41586-024-08449-y
69. https://www.coindesk.com/tech/2025/04/17/quantum-computing-group-offers-1-btc-to-whoever-breaks-bitcoin-s-cryptographic-key
70. https://digital-strategy.ec.europa.eu/en/news/commission-publishes-recommendation-post-quantum-cryptography, last accessed 22/4/2025.
71. https://uk.finance.yahoo.com/news/europe-post-quantum-cryptography-market-151700764.html; last accessed 23/4/2025.
72. https://www.mkm.ee/en/news/estonia-joined-eus-cooperation-framework-quantum-communication-infrastructure; last accessed 28/4/2025.
73. https://neic.no/news/2022/08/19/nordiquest-has-started/; last accessed 28/4/2025.
74. https://sciencebusiness.net/network-updates/university-tartu-and-cybernetica-cooperate-study-quantum-safe-cryptography; last accessed 28/4/2025.
75. https://cordis.europa.eu/project/id/101087529; last accessed 28/4/2025.
76. https://digital-strategy.ec.europa.eu/en/library/recommendation-coordinated-implementation-roadmap-transition-post-quantum-cryptography; last accessed 27/4/2025
77. ZK podcast, "*Lattice based ZK Systems with Vadim Lyubashevsky*", Episode 359, 30th April 2025, https://youtu.be/yQD65PhFwwY
78. https://www.enisa.europa.eu/sites/default/files/2024-11/Remote%20ID%20Proofing%20Good%20Practices_en_0.pdf; last accessed 22/4/2025.
79. https://e-estonia.com/cybernetica-post-quantum-cryptography-joins-to-menu/; last accessed 28/4/2025.
80. Temporarily entailed at the end of the paper for now to maintain anonymization
81. https://medium.com/@shymaa.arafat/what-are-verkle-trees-kzgcommitments-and-could-they-be-applied-on-bitcoin-cbf4838d18ac; last accessed 28/4/2025.

Appendix A: The detailed steps of a complete Verkle Tree (VT) Solution

A.1 Setup:

The election authority picks a random secret τ to be destroyed after the end of the setup procedure and calculates (offline) the following terms with a finite cyclic group

G that is closed over the used finite field [58,81]; the calculated terms should be public to verifiers:
$H_0 = G$, $H_1 = \zeta \cdot G$, $H_2 = \zeta^2 \cdot G$, ..., $H_{p-1} = \zeta^{(p-1)} \cdot G$

A.2 Steps:
-When a new vote enters the system (through the vote collector and the registration applications), it goes through the following steps (Fig.6):

1. When a new vote record enters the vote collector, in addition to adding it to the votes list, the system cryptographically commits to the vote record by adding its hash to the first Verkle Tree; VT_1 = VT_1 + Hash(vote_i)*H_i
2. If this voter-ID hasn't appeared before, the new vote hash is aggregated into the second VT as well; VT_2 = VT_2 + Hash(vote_i)*H_i
3. If it's a repeated voter-ID, then before aggregating it to the second VT, the previous vote must be deleted from it first[30]. The deleted vote is then aggregated into the third VT (or the corresponding second level VT if needed). Since [38] shows that multiple voters are sparse, O(10,000) multiple votes, second level VTs could be implemented as an array holding (voter-ID, VT value)
4. The fact that a voter has voted at the polling station, will not be available during this phase to separate them; if Range proofs were not deployed, then the separation of invalid votes will not be possible at this phase too. However, the Ballot Processor can commit to their values (after separating them as in Fig.6) by calculating VT_4 and VT_5 respectively.

-When i-voting time is closed, make all VT values publicly available in an immutable way; this could be done through the official election site or any trusted robust blockchain.

[30] We assume the index of the old multiple vote make it retrievable from the list of votes, otherwise step3 will be done by the Ballot Processor; also, this could be done in the background without delaying the interactive processing with voters.

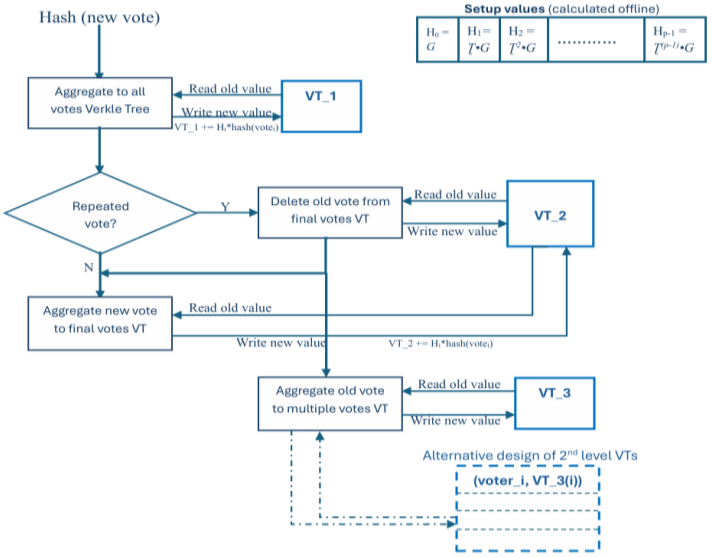

Fig.4. A schematic flowchart of the proposed Verkle Trees design

A.3 Results

The proposed Verkle Trees adds the following security values:

- The polynomial degree of each Verkle Tree proves the number of votes it holds (*n* of a Verkle Tree is cryptographically proved); i.e., two values from Eq.1 in section 5.2: *Count (original votes file)* and *Count (multiple votes).* The Ballot Processor should check that ***VT_1=VT_2 + VT_3***[31].
- In addition to the above check, publishing all the VT commitments allows independent verifiers to write their own code to calculate VT_end of anonymized votes. Then, taking advantage of the Homomorphic property of KZG commitments, verify that ***VT_end = VT_2 – (VT_4 + VT_5)***.
- If RLAs were applied and samples were selected to verify manually, the calculated VT values can also provide separate succinct proof for every sampled value; this is possible whether samples were taken as ballots or as voters. The alternative design of 2nd level VT_3 (Fig.7) may facilitate individual voter verification in a more detailed way; VT_3(i) can prove the number and values of every deleted (multiple) vote for the sampled voters, even if they voted at a poll station afterwards.
- In the 2-levels VT_3 case, **QR codes can include the number of multiple votes** for each voter; this is an extra check against vote manipulation[32].
- **VT**_s could be kept after destroying election data to verify individual votes.

[31] Addition here means multiplying the two polynomials holding the data, where the resulting polynomial degree (number of committed elements) will be the sum of their degrees.

[32] There is a maximum of 3 QR code checks, but a voter could vote 10 times and check only the QR of the last vote; in this case our modified QR will reply that "*you voted 10 times, your last vote is…*".

[80] A Brief on Post Quantum Online Voting and Replacements of El-Gamal Encryption

1 Companies

We have discussed Cybernetica in the main text; [79] explains that they have different countries as customers some of which have 2030 PCQ migration plans, while Estonia is expected to be sooner than that. Another software company, Wisekey located in Switzerland [81], offers its post quantum e-voting services with AI and blockchains as extra lines of defense; however, we did not encounter any real deployment yet.

2 PCQ i-voting

There are few earlier attempts, all of which are lattice-based; [82/sections 4.5&6] presented a comparative summary and found, as of 2022, post quantum e-voting research to be *in its initial stages and has not been fully developed*. Also Cybernetica-Tartu paper in 2024 [59] identified "*research lacking on PQC for esoteric use cases such as i-voting*" as one of the obstacles.

Exploring few existing papers, we find [83] from 2016 relied on the honesty of the Bulletin Board and was inefficient for depending on fully homomorphic encryption without ZKPs, then in 2017 [84] deployed ZKPs and proof batching to be more efficient but still the number of needed proofs increases logarithmically with the number of candidates, while [85] (2021) overweighs the election authority trust assumption which is highly unrecommended in the Estonian case. Another 2024 paper [86] that applies Lamport signatures on Merkle data (to decrease the data size); although seems promising, the Merkle tree hash-based signature grows with the number of leaves linearly in the private key size and logarithmically in the signature size (with large constant as well, 256*log n)[33]. Finally, we mention [87] from 2025 which is a lattice-based e-voting protocol following the standard mix-and-decrypt framework and supports general ballots.

2 PCQ El-Gamal replacements

Although dated in 2020, [60] presented a straightforward solution For El-Gamal based systems where the user(voter)'s public key is encrypted by a symmetric encryption algorithm (Blowfish in [60] but could be AES for more security) and then shared using *QKD* (*Quantum Key Distribution*); the same quantum key is used for the symmetric encryption. In this arrangement the resulting public key can only be used once, which may complicate multiple voting; the future work of [60] included the use of *S13 QKD* which can perform the encryption and public key sharing in one step, then they introduced further enhancement, [88] on S13 in 2023.

Another newer (2024) alternative for El-Gamal replacement that aligns better with ENISA recommendations and NIST final standards is introduced in [63]; a lattice-based scheme that depends on the hardness of the *SIS* problem and is coupled with a private-key encryption scheme. Note that the encryption schemes used in both cases must be secure against *Chosen Ciphertext Attack (CCA)*.

[33] The IBM researcher mentioned in [77] (30/4/2025) that there is new Lattice-based research (accepted to appear soon) that produces ZK proofs and/or signatures with size O(log log n).

References of [80]

82. https://www.wisekey.com/press/wisekey-pki-and-sealsq-post-quantum-technologies-enhance-e-voting-security-through-advanced-cybersecurity-and-ai-integration/
83. I. Chillotti, N.s Gama, M. Georgieva, and M. Izabach`ene. "*A homomorphic LWE based e-voting scheme*", 2016, In PQCrypto, volume 9606 of LNCS, pages 245–265. Springer
84. Rafa¨el del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler, "*Practical quantum-safe voting from lattices*", 2017, In ACM Conference on Computer and Communications Security, pages 1565–1581. ACM.
85. Guillaume Kaim, Sébastien Canard, Adeline Roux-Langlois, Jacques Traoré, "*Post-quantum Online Voting Scheme*", FC 2021 - Financial Cryptography and Data Security. International Workshops, Mar 2021, Virtual event, France. pp.290-305, DOI:10.1007/978-3-662-63958-0_25 ;https://hal.science/hal-03355875
86. https://www.researchgate.net/publication/360161016_A_Review_of_Cryptographic_Electronic_Voting
87. S. Shriya, J. D. Sweetlin and V. Supraja, *"Secure Online Voting System Using Hybrid Post-Quantum Signatures,"* 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2024, pp. 1-7, doi: 10.1109/ACCAI61061.2024.10602388.
88. Hough, Sandsbråten, and Silde , "*More Efficient Lattice-Based Electronic Voting*", 13-Jan-2025, from NTRUIACR Communications in Cryptology10.62056/a69qudhdj**1**:4; https://cic.iacr.org/p/1/4/10/pdf
89. Bello A. Buhari, Afolayan A. Obiniyi, Sahalu Junaidu, and A.F.D. Kana, "*Enhancement of S13 Quantum Key Distribution Protocol by Employing Polarization Secrete Key Disclosure and Non-repudiation*", Aug 2023, International Journal of Wireless and Microwave Technologies, https://www.researchgate.net/publication/372769399_Enhancement_of_S13_Quantum_Key_Distribution_Protocol_by_Employing_Polarization_Secrete_Key_Disclosure_and_Non-repudiation