(**a preprint version to be submitted to preprint servers**) by
Shymaa M. Arafat
Associate Professor in Computer Science & Engineering (Independent)
Shar.academic@gmail.com, Shymaa.Arafat@gmail.com

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 academic literature 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 complaints, and about newer solutions suggested by academia in 2024. Finally, we discuss the current system status in 2024 EP elections and propose our own suggestions to some problems stated in the *OSCE-ODIHR* 2023 report that are still there.
Keywords: IVXV, El-Gamal Encryption, Verkle Trees, vote buying, counted-as-casted.

1 Introduction

Estonia is a small 1.37m 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***[3] June **2023** report [2].

Although the ratio of citizens preferring i-voting is continuously increasing and has reached 51% in 2023, [3], one can trace a long history of rejection incidences from conservative right parties in [1,2 page8 footnotes16&17] from 2005 up till now; the

[1] Although we used [1] as reference for the whole sentence, we wrote the updated number 1.374687m as of Jan 2024 from (https://news.err.ee/1609341741/statistics-estonia-s-2024-population-more-than-1-37-million) which mentions too that most of the annual increase (9000) are Ukrainians. More detailed statistics can be found in (https://www.stat.ee/en/find-statistics/statistics-theme/population/population-figure#) which tell us that there are **1.127m "citizens"** as of Aug 2024 and the rest of the population are just residents; it also mentions that there are 296,268 of the citizens **~ 26.28% are from Russian ethnic** nationality.

[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.

[3] *OSCE* stands for Organization for Security and Cooperation in Europe, *ODHIR* stands for Office for Democratic Institutions and Human Rights; for IVXV to be used in voting for the European Parliament elections, European entities ought to approve its security.

situation was emphasized in 2023 when the internet votes flipped the election result for one of those parties, *EKRE*. The distribution of internet votes was completely different than that for poll station paper votes as shown in the curves in [4]; analysts due that to the society division mentioned above (i.e., it's expected for the distribution to reflect the parties 'conventions on the voting method). Still, there were some complaining movements that continued persisting to the 2024 European elections [5].

Since most rejections come from right parties, a note about the effect of Ukrainian war on Estonia being the closest neighbor is appropriate; according to [6] there were Russian attacks on the system, but the authorities say it was properly defended[4]. On the political impact, we suffice with this article [7]; even if it was politically biased, it points out to the economic status with more refugees that the writer projects will seek Estonian citizenship, and thus can vote. Also, some parts in the context of the *OSCE* report, [2, pages 15-16, footnotes 45&48], implies that Estonia is not very strict in giving citizenship for stable residents and their born children[5]. The paper does not aim to dig deeper into political details; however, it is important to enlighten the reader about what seems to be the roots behind the split of opinions in the Estonian society.

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 with important improvements that were done or proposed in 2023/2024, and section 5 discusses the remaining vulnerabilities of the system along with suggested solutions by the authors. Finally, section 6 introduces suggestions for further future research and section 7 concludes the paper.

2 Recent opposing activities with technical merit

A technical incident that gained some publicity in 2023 elections [8] was done by the same computer scientist observer[6] in [4]; he voted using his own Python code [9,10], meaning that he *has overridden the official voting application* that voters

[4] We clarify that although [6] 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.

[5] It is not of our concern whether OSCE finds this fact good or bad; we are just stating the facts. Also, we decided to keep the article [7] because we believe it is our obligation as an SoK paper to inform the reader with the whole picture even if it involves some extremes.

[6] 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). On the other hand, the term "***Computer Scientist***" taken from the title of [4] is extremely rejected by IVXV representatives and supporters who describe Mart Poder as "***A hobby hacker activist***"; the OSCE report referred to him as "***someone with sufficient programming skills***"

should download to deliver their vote to the system. This gives an alarm that the voting application is not authenticated by the system, which was noticed by the *OSCE* report [2, page 8][7]; there were a mention of *some wrong district votes* too [11], but the report says they were corrected except one vote. 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 get handled in the following election (following footnotes, 14&15, contributes to this impression). A recent complaint about *the decryption of invalid votes* after the current 2024 European elections (events are happening while we are writing) was also rejected objectively in [12]. Among the three listed reasons, being an *observer* not an *auditor* seems to be the dominant one, where auditing is organized by the State Electoral Office in all elections [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; however, *the file containing the decryption of invalid votes is only accessible to auditors* [14, page 22].

The recent European parliament elections has brought some newer actions from the i-voting opposing community [5,12,15,16,17]; the same observer mentioned above has developed some kind of shadow e-voting site they call *virtual threshold survey* [15] encouraging citizens[8] to vote again on it as a check. Also, a criticizing paragraph about IVXV strategic mistakes from a researcher in *Cybernetics* TUT is being circulated online [16][9]; however, a clarifying response told us 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* which is the company behind the current Estonian internet voting system since 2014 (partnering with -*Smartmatic* [3,18]) as we will detail in the following sections. However, in addition to the complaint in [12], we find the story of an earlier complaint about the election desktop also alarming; a first complaint granted the observer (on 23/2/2023) a 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 [19]. The observer took the

[7] We received some rejecting opinions that it is considered a system strength and a transparency quality that voters can vote using their own applications; however, since the OSCE report described it as "***could present a cyber security risk***", and even the IVXV team did not acknowledge this point of view when we mentioned it in [40] as a possible reason behind not authenticating the voting application, we stick to what is detailed in section 5.1 (and [40])

[8] We have no evidence of considerable participation ratio till the time of writing.

[9] The statement in [16] about "*a strategic mistake*" was made on March 2024, he made another (more moderate) statement on June (https://arvamus.postimees.ee/8036445/ago-samoson-selline-kontroll-parandaks-e-valimiste-usaldusvaarsust). People from the system says that although AGO Samoson is a 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).

photo shown in Fig.1 when the disk inspection took place (28/11/2023)[10]; 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*"

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 Windows 10 operating system, but the operator didn't even remember that DigiDoc4, Notepad++ and RamDisk tools are also installed on it. From [19], we say maybe it is just a backup PC (not the one used in key creation)

Finally, we haven't seen yet an *OSCE-ODIHR* report on the 2024 European elections, but another scientific report from ***the cyber security committee of the Academy of Sciences*** has been already handed to the election organizers [20]. Although the report is still under review and are not published yet, Google translating the minutes of the last committee meeting [21] on 3rd of June tell us they *have identified **6 threats** whose risk class is higher than small*.[11]

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, Fig.2, 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 [22]; 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.

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

[11] 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).

Fig.2: Estonian eID card with 2 keys (authentication & signature), image taken from [1]

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, Fig.3 is a Table taken form [23].

Certificate pairs with duplicate RSA public keys

No	Time of cert issuance	Type	Cardholder	Issuance	Expiry date	Revoked	Warranty
1	2012-11-06 15:35:09	sign	Ulle	PPA renewal	2016-07-07	2016-06-27	2014-10-09
	2012-11-06 15:35:46	auth	Toivo	PPA renewal	2016-07-04	2014-11-21	2014-10-09
2	2013-02-06 15:35:54	auth	Phillip	PPA renewal	2016-11-14	2015-05-04	2015-01-06
	2013-02-06 15:35:56	sign					
3	2013-02-07 12:18:34	auth	Sandra	PPA renewal	2016-01-02	expired	not issued
	2013-02-07 12:18:37	sign					
4	2013-02-19 09:09:58	auth	Nadiia	PPA renewal	2016-11-24	2016-11-08	2014-12-22
	2013-02-19 09:10:08	sign					
5	2013-02-25 09:33:17	auth	Moonika	PPA renewal	2016-08-22	2014-12-30	2014-12-22
	2013-02-25 09:33:29	sign					
6	2013-03-04 11:36:08	sign	Richard	PPA renewal	2016-11-30	2014-10-13	2014-10-09
	2013-03-04 11:36:38	auth	Anu	PPA renewal	2016-08-12	2014-10-23	2014-10-09
7	2013-03-30 13:40:38	auth	Leili	initial	2018-03-26	2015-05-14	2014-12-22
	2013-03-30 13:40:40	sign					
8	2013-03-30 13:42:03	auth	Jaan	initial	2018-03-26	2014-12-30	2014-12-22
	2013-03-30 13:42:05	sign					
9	2013-04-15 09:16:11	auth	Liis	PPA renewal	2016-05-06	expired	2014-12-22
	2013-04-15 09:16:28	sign					
10	2014-10-08 12:01:16	auth	Siim	initial	2019-10-07	2017-10-03	not issued
	2014-10-08 12:04:31	sign					

Fig.3: First incidences of duplicate RSA keys whose owners were told to renew their ID cards at PPA stations; the table adopted from [23] with a detailed analysis of the marked with red case (they proved a valid signature for the first using the keys of the second)

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

[23] and the paper itself [24]; more faulty keys issues[12] can be found in the PhD of the same researcher Arnis Parsovs [25], and in [26]. Also, other RSA vulnerabilities were discovered in [27].
According to [1], this was a global crisis for the company which was sued in many other countries around the world; Spain and Slovakia [28] replaced all the physical cards while Estonia fixed them remotely. Then, they changed the company to ***IDEMIA*** [29] and [23] recommended moving to threshold cryptography and homomorphic encryption; the Estonian i-voting system IVXV uses ***384-bit Elliptic Curve*** Cryptography ECC and El-Gamal Encryption[13].

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 was there since 2005. Then, based on [sec.1 of 30,31,32], we mark 2 milestone step transitions:

- In 2011, a student named *Paavo Pihelgas*[14] 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. However, several flaws were discovered in 2014-2016 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 in IVXV[15].
- 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, the 2017 and

[12] 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.

[13] According to the official documents (page 14 in [33]) the "authorized voters list" is still signed using an 2048 bit RSA key.

[14] According to [32] 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)

[15] The verifying extension, was only added in 2013 based on S. Heiberg & J. Willemson suggestion (https://ieeexplore.ieee.org/document/7001135), possible attacks were discovered in (https://dl.acm.org/doi/10.1145/2660267.2660315) and then the verification was improved on 2016 (https://research.cyber.ee/~janwil/publ/ivxv-evoteid.pdf)

current version of 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 [33]; we also detail two enhancements that happened in 2023/2024 before getting into the vulnerabilities of the current systems status in section 5.

4.1 Brief Factsheet

For a factsheet summary, the developing companies are ***Cybernetica-Smartmatic***; the voting device must be a desktop PC (mobile voting is still postponed at least to 2025 [34]); voting on the other hand can be done using *mobile-ID*, *Smart-ID*, or any digital identity integrated in the *web-eID*[16]; *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.4; the voting steps could be summarized as follows

[16] The web-eID solution enables the use of different digital identities available in Estonia, including ID-card and digital ID, (https://www.id.ee/en/article/web-eid/); this is part of applying the European Union web-eID project for all public key cryptography digital identities across Europe (https://github.com/web-eid/web-eid-system-architecture-doc). 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 in https://www.valimised.ee/sites/default/files/2024-05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-arhitektuur%29.pdf, and secs 8.5-8.6 of the corresponding protocols file; both are available only in Estonian language, but translatable by web browsers).

1. The voter installs the voting application[17] , 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 application*, **RA**, and if eligible displays the candidate choices for that voter (according to district if it's a local election).

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 *timestamp certificate*[18] 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[19], then decrypted at the counting phase; the ballot processor removes multiple votes after voting is closed, and before sending to mix-nets.

5. The vote collector sends a verifying ***QR code***[20] to the voter for optional vote verifying (through verification application) using a second smart device.

---

[17] Sometimes called the voter application in official documents, but we prefer to follow the naming convention in [31] to make it clearly distinguishable from the voter device.

[18] Before, the timestamp certificate was important to distinguish the last vote of each voter, and also for checking the possible verify duration of 30 mins; in the newer version *the certificate* sent by the Registration to the Collector *is* a ***signed CONFIRMATION*** (by RA) that contains the original request (now called ***ORDER***) sent (and signed) by the VC, along with the *timestamp*. The VA should check both signatures in the certificate it receives.

[19] 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 [30])

[20] According to [9], there were also a revealing incident of *the president vote* through his QR code (when i-voted live in front of cameras, and showing his QR code, to encourage citizens to vote online; people took a snapshot of the QR code and revealed his vote). The incident was mentioned in the context of doubting privacy and protection from coercion and/or vote buying.

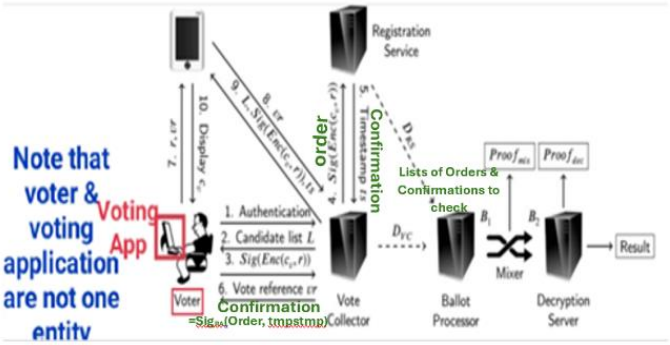

Fig.4: a diagram describing the architecture & the steps of the Estonian voting system, adopted from [1] (we added some colored remarks to remind the reader to distinguish between the voter machine and the voting application when it comes to attacks & vulnerabilities; we also added updates in IVXV 2024 version 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 [31]; Fig.5 omits the voter signature and mix-nets parts and concentrate on the parts involved in the exploit and the introduced possible attacks.

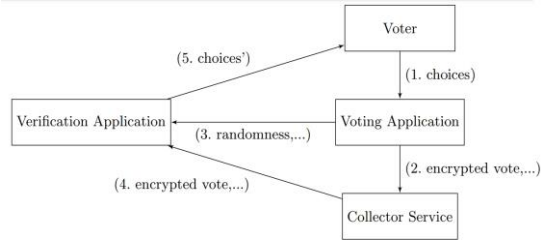

Fig.5: the part of vote casting and verification taken from [31]

Let the election public key be "**y**" with corresponding secret key "$\mathbf{S_k}$", and "**g**" be the generator for El-Gamal encryption[21]; 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.

[21] We've mentioned in section 3.1 that a 384-bit elliptic Curve is used for the group field.

-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 C_2 from the vote collector; this gives a malicious voting application the chance to manipulate the encrypted cipher text by *sending wrong "r" value* to deceive the verification application (different values of C_1 for the same C_2).
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 [35, lines 77-83 & 141-146 in code and the exception is thrown at line 60] on Feb 2023 just before March 2023 elections[22]. The authors alarmed that it is concerning [31, sec 3.6] that such a straightforward vulnerability wasn't noticed earlier, and then criticized the quality of IVXV in general [31, sec.4].

4.4 Range Proofs & Invalid Votes

Another improvement was added to IVXV on 30th May 2024 [36][23], 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; that's why [14] suggested preventing invalid ballots from reaching the decryption phase at all. The thesis suggested the use of range proofs (based on *Bullet Proofs* & *Pederson Commitments*)[24] by the vote collector application to check the validity of the vote it receives from the voting application without revealing it; hence, the vote collector would be able to reject invalid votes and do not add them to the list of votes. The thesis preferred the use of ***Range Proofs*** as opposed to ***Set-Membership***

[22] One may find it suspicious that they didn't fix it when the authors of [31] first sent them a letter and waited till the last month before the election; clearly, the fix was after the time of writing because the authors stated: "*up till now the vulnerability was not fixed"* and that whenever they asked the reply was "*we're working on it*".

[23] The GitHub in [36] shows that this version of IVXV is named 1.9.10-EP2024, while the previous fixed version in Feb 2023, [18], was named 1.8.2-RK2023.

[24] The authors of [14], and Microsoft electionguard as we will point out later in the paper, preferred those as they are based on the *Discrete Logarithm problem* like El-Gamal encryption and considered general purpose SNARKs based on polynomial commitments as not suitable for El-Gamal based voting systems, while a more recent paper (a follow up to *Kryvos*, the one [14] references as [37]) introduced benchmarks for an efficient implementation of *Groth16* over them (https://link.springer.com/chapter/10.1007/978-3-031-72244-8_7); another 2024 paper (https://link.springer.com/chapter/10.1007/978-3-031-68403-6_6) introduces *Polymath* proving it can be more efficient than *Groth16* and relates it to KZG commitments. While comparing those approaches can be a rich research area, some readers might find it out the scope of this paper; Hence, a condensed summary of variations between possible SNARK choices in the first 25 mins of (https://youtu.be/A3edAQDPnDY) can be enough to fulfill their curiosity seeking a wider vision of the subject.

proofs for their simplicity and suggested some mitigations to the discontinuity of the set of vote choices.

Why not reveal invalid votes?
As mentioned in section 2, there was a lot of debate and complaining 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 match accurately those explained 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 [36]. 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 [37, 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 [38] to be ***exactly 1*** in the last three elections *since 2021* make it look quite suspicious; the doubt includes anyone who can see the votes.

5 Remaining Vulnerabilities/Issues

We do not aim to diminish the long Estonian experiment in i-voting that approaches two decades, but there are problems. Although there is not enough documentation yet to accurately trace all changes in the newer better version, IVXV 1.9.10-EP2024[25], we know it is the one analyzed in the most recent report [21] which identified 6 threats with risk level higher than small. The report was done on collaboration with the election authorities (i.e. not biased against i-voting); the *OSCE* 2023 report [2] pointed out to some issues too we have no evidence they're resolved.

As for academic research papers, in addition to pointing out attacks [31,32,37], many have introduced a holistic criticism to IVXV; [31] analyzed *IVXV public information* as *satisfying* only *1* (minimal restriction on disclosure of vulnerabilities)

[25] GitHub history shows 897 changed files with 34,059 additions & 10,830 deletions; translating available pdf files shows the work done in integrating different kinds of digital identities, and in coordinating with the XRoad service (X-tee); also a whole section (6 in the protocols file in [36]) is dedicated to the registration service and its communication with online (RIA), offline (RVT) and other IVXV services.

out of 9 quality metrics[26] and warned from the possible existence of hidden vulnerabilities; [37] 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*; [39, sec. 5.1] discussed the different trust assumptions of IVXV including software components and key holders, in this context [37] also discussed that Vote Collector is trusted on the privacy of encrypted votes.

The IVXV team on the other hand sticks to the claim that there is no proved error in the election results; however, with *QR verification ratio of 5.5-9.9%*[27] as stated in the official i-voting statistics site [38], this does not really prove beyond reasonable doubt that no encryption pairs were manipulated.

In this section we discuss some obvious vulnerabilities in IVXV with possible solutions.

5.1 The Voting Application

The authors of [31] commented on the voting application being the only unrevealed part of IVXV code as an open source, while [39] 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***[28]; 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, a ratio that could even increase by social engineering to target those who are not likely to verify, makes the risk level more severe.

A possible attack by a malicious voting application that could deceive even verifying voters was discovered by Olivier Pereira in [32]; *a malicious application could fake a system crash to* deceive the voter to vote again. This way the application will ***take the voter signature twice*** (generate two votes and two "*r*" values); hence while showing the voter the QR code of his/her choice, the system will consider it an

[26] The quality standards are from an earlier, FC'21, paper by the same authors (*New standards for e-voting systems: Reflections on source code examinations*)

[27] The highest recorded verification ratios are 6.7% in 2021 elections, then when the statistics for the European 2024 elections appeared it showed a significance increase to 9.9%. It's not clear whether the ratio is per voter or per vote (in case of multiple voting); however, considering 2023 elections, since total cancelled multiple votes is 12,119 (10,787 multiple + 1,332 replaced by paper) out of 313,514 total i-votes (~ 3.86% with a max. of 9.27% in 2021 elections), we don't think this will make a significant difference.

[28] Lately, some of IVXV team respond that "*being able to vote with your own-made application is considered as a measure contributing to the transparency of the system in Estonia*"; although one do not get that impression from their previous email response in [40], the solutions we suggest can still provide this feature.

old vote. Although the author suggested few mitigations, we have no clue that any of them was adopted, except storing a voteID field with each vote although we are not sure if it was added to the QR code. We 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 (https://myid.skidsolutions.eu/en), especially if their device suffered a system crash while voting; would also be helpful if the QR code contained a counter on the number of multiple votes. Another simple safeguard for this specific problem only, [40], would be *to force a time interval between votes; the verification interval, 30 mins, seems a suitable choice*[29]. However, this must be accompanied by heavily warning voters to close the application then reopen again after 30 mins; if the voter eID remained available on the voter's PC more than 30 mins, a *Ghost Click attack* becomes possible [32] and a malicious voting application would have enough time to submit silently without voter's knowledge.

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[30] before transferring the money. This *DarkDAO*s idea was discovered by [41] 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 [42] with a GitHub code[31]. Also, another group of researchers have recently discussed in [43] the newly evolved threat of using new technologies like TEEs and blockchains by coercers or vote buyers.

A general solution to all the above would be ***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[32];

[29] The authors of this paper sent a few suggestions to information systems emails from (https://www.valimised.ee/index.php/en/electoral-organizers/state-electoral-office/staff), and the time interval is the only one they considered "possible" in their reply. An appendix is added with the full emails (to be replaced with a reference, [40], to a PDF file when the authors names are revealed).

[30] Remote execution attests were originally discussed on Intel SGX (which is available on many new PCs in the market: https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions-processors.html), and it exists in other processor companies like Apple and even mobiles as well. So, we can assume there's a considerable probability that the voter PC can support it.

[31] The same authors simultaneously participated in another paper which suggests a solution that TEE provides only execution attests after submitting ***proof of Complete Knowledge (CK***); however, they were only concerned with TEE mobile devices (https://medium.com/initc3org/complete-knowledge-eecdda172a81, https://eprint.iacr.org/2023/044)

[32] The Electrum cryptocurrency wallet already does that through PGP (https://bitcoinelectrum.com/how-to-verify-your-electrum-download/, https://en.bitcoin.it/wiki/Electrum#:~:text=8%20References-

this is very much voter dependent but still gives 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 is to assign *a signature & authentication key pair* to the voting application like the rest of involved 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, [40], *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 like a *PC interrupt*) that pops up on the voter screen. The use of *a feedback channel* was also one of the suggested mitigations in Olivier Pereira paper [32] 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 involved in implementing this solution, it is feasible to implement[33], and they already acquire knowledge of the IP address to calculate ratio of abroad voters as stated in [38].

5.2 Cryptographic Proofs

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***". Again, this was viewed in [39] as trusting the vote collector and registration

,Verifying%20Electrum%20Binaries,files%20were%20not%20tampered%20with.). Hence, it is feasible to do so.

[33] In (https://arxiv.org/abs/2411.11796) the authors mention (page 3 section 2) that the Kiosk in CAC-Vote can identify the voting machine although not explained how; digging further, the last few lines in the question (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). In any case, any other application in the Estonian e-government that links electronic IDs to cellular numbers or emails can receive just the voter ID to send a simple fixed line "*Beware you are not using the official voting application*".

applications to not collude[34], otherwise it would be possible to drop ballots; recall also the threats in [37] since they can view the decryption of invalid ballots.

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 [30], 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*. This was integrated in the audit application as the ***Integrity check*** in [44], published on 30/5/2024, just before the European parliament election[35].

-A possible general *double check* for the whole list of votes is to compare (using ZKPs) with the transaction records of the Estonian Information System, [40], like checking that *the total number of transactions to IVXV services equal the total number of existing ballots*[36]. If this is not possible to be done for all votes, it could be done on sample of randomly selected votes as a form of *Risk Limiting Audits* ***RLA*****s**, where the transactions could play the role of paper ballots in paper dual e-voting systems.
-We also think the new version [36, sec.6] might have added some comparisons in that direction with *X-tree* service of *XRoad*. They have also included some handling within RFC 3161; ***PKIX*** is a timestamping protocol where a trusted third party can confirm the existence of data at a specific point in time with its signature.

-**The *Range Proofs*** solution [14] to the problem of *invalid votes* was discussed in section 4.4; we should also mention that [45] assumed the existence of NIZK Range Proof for ballot correctness in their system that is implemented according to *ElectionGuard* v.2 specifications, and when traced their code we found it uses a

[34] Although Jan Willemson stated in his 2022 paper, [30], that the registration service makes it "*impossible"* to drop ballots, more tight statement about the trust assumptions involved in his newer (2023) co-authored paper [39]; we also have mentioned before that this is much more clarified (with figures) in the newer IVXV protocols file in [36].

[35] A newer paper (https://ieeexplore.ieee.org/document/10811882/) appeared in Dec 2024, after this paper was written, showing that the authors' suggested integrity checks were added to IVXV before the European parliament election, [44], to detect insider risks at the processing stage. The checks include comparing checksums of subtotals, SHA256 hashes of totals, subtotals and individual votes; also, *count-based validation* is used to detect manipulations like adding the removed older multiple votes to the list of anonymized votes before getting into the mix-nets. However, as shown in [44, line 239], votes are stored in a ***Treebag*** which represents a binary search tree data structure in Java; i.e., resembles a Merkle tree where the number of nodes is not cryptographically verified as in the case of Verkle tree we are going to discuss in the following subsection 5.3

[36] In light of [43], double checking multiple votes with the *KSI blockchain* might be another area where adversaries might do what the authorities haven't done to catch voters who voted again after their coerced vote; further investigation is needed to investigate such possibility and other risks their work may alarm about.

disjunctive Chaum-Pederson Range Proof[37]. Thus, we concentrate on proving the removal of multiple votes (and not dropping any votes) in the following sub section.

5.3 The proposed Verkle Tree

-For this problem *we suggest to aggregate all vote hashes in a 2-level Authenticated Data Structure ADS* [46, sec.2.1 def.3] which we can simply describe as data structures that can provide a succinct (short) cryptographic proof (sometimes called witness) of each element stored in them. In fact, we think this could be similar to what was suggested in [37] as a protection from privacy attacks; they suggested each voter computes a *NIZKP of knowledge* of his/her encrypted vote, and we suggest the system stores all votes in an ADS that can later generate all such proofs.

In this paper, we choose ***Verkle Trees*** [47], which is a vector data structure that authenticate its elements based on KZG polynomial commitments because they have the shortest verifying complexity (constant order), and because they provide a cryptographic proof of the number of elements stored in them (as opposed to Merkle Trees) which is beneficial in our case. Hence, we propose to aggregate all votes in a Verkle Tree, such that votes from the same voter are aggregated in a second level tree. The second level tree could either be a Merkle Tree[38] with the multiple vote counter also hashed inside the verkle node (to be cryptographically verified) as in Fig.6, or could also be a verkle tree to verify the multiple vote count.

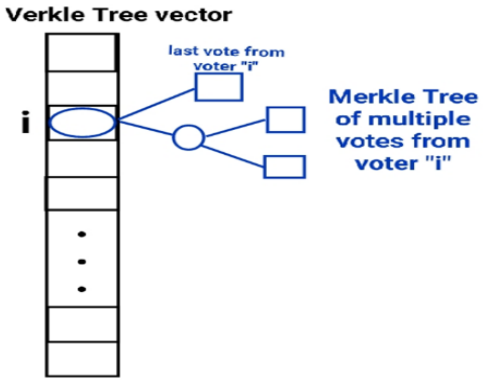

Fig.6: two-level Authenticated Data Structure, a vector that will be committed to using KZG commitments as a Verkle Tree, where each node value could be the root of a Merkle Tree containing multiple votes from the same voter

[37] https://github.com/microsoft/electionguard-rust/blob/main/TODO.txt#L1373; recall that we have already mentioned in footnote "24" other approaches suggested in the literature (like Groth16 in https://link.springer.com/chapter/10.1007/978-3-031-72244-8_7).

[38] We use Merkle (not Verkle) Tree in the second level, because we expect (in light of the available statistics) no voter will vote more than 8-16 times; ie, those subtrees rarely exists and are of 3-4 levels at maximum.

To clarify more on the details:

-When a new vote enters the system (through the vote collector and the registration applications):

- If the attached voter signature hasn't appeared before, the new vote is added to the votes list and its hash is aggregated to the Verkle Tree vector commitment.
- If it's a repeated signature, the previous vote is inserted to the corresponding second level Tree attached to the corresponding Verkle node, the second level counter is verified to be incremented by 1, and then the Verkle node is replaced by the new value (the old node is deleted, and the new node is inserted; the counter is incremented).
- If the voter voted at the polling station, its node should still be kept for auditing/observation purposes; whether as a zero value with certain flag or in a separate Verkle Tree this could be an implementation decision.

-At the end:

- ***the number of nodes in the Verkle Tree proves the number of voters who voted (counted votes)*** (*n* of a Verkle Tree is cryptographically proved).
- The counted votes are presented in the Verkle Tree nodes[39] (could be zero if voted at poll station) and their aggregation can be verified cryptographically.
- *Every deleted vote can be traced through its corresponding Merkle/verkle proof*, and their number is also cryptographically verified.
- This way, the **QR code can include the number of multiple votes** for each voter, and possibly lead in the verification to a list of all previous multiple votes for that voter; this is an extra check against vote manipulation[40].
- Whatever the designer decision for handling ill-formed votes and those who voted at polling stations, the point is that ***old votes can be still traced*** and *if the code is open source all the numbers can be cryptographically verified*.

Discussion

In general, this could be viewed as a way to prove that votes are *counted as collected* and remove the trust from some software services.

[39] A developer may choose to store the last vote twice (in the Verkle node and as the last leaf node in the Merkle tree); at this step of analysis, we believe it is an implementation decision.

[40] We know 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…, and your previous votes are". In fact, [9] shows, and demonstrated to the authors through X conversation (https://myid.skidsolutions.eu/en, https://x.com/trtram/status/1763936733027049606), that a sophisticated user can already do something similar through e-ID transaction confirmation service available in Estonia (*myID*); however, our suggestion doesn't contribute to vote coercion since the voter can hide the latter QR code from the coercer.

We chose Verkle Trees since they provide a constant order complexity *SNARK* (per node) using KZG vector polynomial commitments, while traditional widely used Merkle Trees provides logarithmic complexity proofs on the data stored in their leaves; both Verkle and Merkle require trusted setup procedure to generate a crs (*common reference string)*, but we do not consider this a problem since IVXV already include a setup & key generation phase. On the other hand, STARKs (*Scalable Transparent ARguments of Knowledge*) option introduced in [46] does not require a trusted setup phase and provides post-quantum security guarantee at the cost of having sublinear complexity; although it is not post-quantum to begin with, the authors showed a projected performance analysis for applying their approach to EL-Gamal based e-voting systems (but left the impact of mix-nets as a future work), a similar and comparative analysis of the Verkle Tree performance[41] on the IVXV case is a possible future work.

Another interesting intuition from [46], that can work for our Verkle Tree as well, is to *make the verifier checks the generated proof* (in their case with a Merkle root of all votes) rather than original evidence. This can be useful for the IVXV case where election data gets destroyed after a month from the election; the smaller size generated proofs could be kept for late further checks.

6 Possible Future Research Directions

First, more investigation and analysis of possible options and implementations to the idea of keeping some archiving proofs after destroying the election data could be useful to the Estonian system.

Then, on the academic research level, a possible future work is to conduct a comparative analysis between possible zero knowledge proofs that could be used; this includes comparing to the STARK approach suggested in [46] for Homomorphic e-voting schemes, and the somewhat contradicting opinion introduced in [14] that considered SNARKs based on polynomial commitments (like the one proposed in this paper) not a suitable fit for IVXV as it is based on the discrete logarithm problem. The paper in [48] could be viewed as the start of such research thread that went in depth into circuit implementation details to enhance a benchmark performance of a *Groth16* based ballot validity check; they conducted their benchmarks on a 254-bit common elliptic curve BN254 which opens the door to investigating the efficiency when using the 384-bit curve used in IVXV. In addition, the idea is extendable to give another possible future research of seeking efficiency for the suggestions in section 5.3, concerning implementation details (as in [48]), and also theoretical mathematical details and proof batching reductions like the work in [49] which did compare KZG and Groth16 over a 381-bit curve (BLS12-381).

[41] As a rough estimate on practicality, Verkle Trees are used in the Ethereum blockchain which is known to be fast in block production rate and heavy in number of transactions (the authors of this paper are aware they are planning to replace it or pivot it in a binary search tree in the future Beam Chain, but this is for escalating post-quantum protection).

Finally, with the intuition of [43], it could be an interesting future 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.

7 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. Last but not least, we discussed remaining risks and unfixed vulnerabilities; mainly, authenticating the voting application and cryptographically proving the removal of "*not to be counted*" votes step. We suggested some possible solutions including checking the fingerprint or digital signature of the voting application and interrupting the voting process when it is wrong; enforcing a time delay between multiple votes to avoid fake crash attacks; also, double checking the total number of ballots in IVXV with the total number of digital identities transactions to IVXV through the Estonian Information System. We also updated the paper to include a briefing of the recently implemented solution as it appeared in [50] before we introduce our suggested solution of using a 2-level authenticated data structure: ***a*** Verkle Tree to aggregate votes, where ***multiple votes from the same voter are to be accumulated in a Verkle or Merkle Tree whose root is a node in the Verkle commitment***. Finally, we introduced some possible research directions that could evolve from all the introduced and presented material in this paper.

Declaration Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

I'm grateful to everyone participated in making me acquire this level of knowledge & research skills: from my graduation faculty Alexandria Computer Engineering department 30 yrs ago (the encouragement of *Prof. Nagwa Elmekky*, and the guiding soul of the belated *Prof. Ahmed Belal*), to everyone who has put their work free online (Berkeley ZKP MOOC, HAL, iacr, arXiv, E-Vote-ID, Tallinn University). Also, a considerable part of the material presented here about observers came from *Märt*

Põdera's blog, X (previously twitter) posts and conversations leading to more accounts & personnels.

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://gafgaf.infoaed.ee/en/posts/great-divide-in-evoting/; last accessed 14/3/2024.
5. https://ausadvalimised.ee/docs/yhisavaldus2023/; and newer petitions in 2024: https://ausadvalimised.ee/ei-lepi-vaadeldamatusega/, last accessed 13/6/2024; https://x.com/ausadvalimised/status/1808854585597108552, last accessed 5/7/2024.
6. https://electionsnovascotia.ca/PictouWestByElectionEBallot, last accessed 22/4/2024.
7. Stone Bridge, "*About the destruction of Eestlus on the example of the Central Party*", https://uueduudised.ee/arvamus/kivisildnik-eestluse-havingust-keskerakonna-naitel/; last accessed 14/1/2024.
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.
17. "*E-voting system Disk appeared out of nowhere*", https://gafgaf.infoaed.ee/posts/esoteeriline-turvamudel/; last accessed 22/5/2024.
18. Smartmatic-Cybernetica. IVXV Voting Service. Version 1.8.2-RK2023, https://github.com/valimised/ivxv/tree/master
19. 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.

20. The report of *the cyber security committee of the Academy of Sciences*, https://x.com/danbogdanov/status/1802998209649762582; last accessed 6/7/2024.
21. Cyber Security Commission minutes of meetings, https://www.akadeemia.ee/akadeemia/noukogud-ja-komisjonid/kuberturvalisuse-komisjon/; last accessed 7/7/2024.
22. 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.
23. Arnis Parsovs, "*Estonian Electronic Identity Card: Security Flaws in Key Management*"; video https://www.usenix.org/conference/usenixsecurity20/presentation/parsovs
24. Arnis Parsovs, "*Estonian Electronic Identity Card: Security Flaws in Key Management*", 29th USENIX Security Symposium, Aug 2020, 978-1-939133-17-5.
25. Arnis Parsovs, "Estonian Electronic Identity Card and its Security Challenges", PhD Thesis, University of Tartu.
26. 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.
27. 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
28. https://e-estonia.com/raulwalter-estonia-digital-identity-giant/; last accessed 20/3/2024
29. https://e-estonia.com/estonia-introduced-a-new-id-card/; last accessed 20/3/2024.
30. 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
31. 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
32. Olivier Pereira, https://www.researchgate.net/publication/372570425_Individual_Verifiability_and_Revoting_in_the_Estonian_Internet_Voting_System
33. https://valimised.ee/sites/default/files/2023-02/IVXV-protocols.pdf; newer, IVXV 1.9.10, in Est language: https://www.valimised.ee/sites/default/files/2024-05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-arhitektuur%29.pdf
34. https://news.err.ee/1609194064/mobile-voting-likely-to-arrive-in-estonia-in-2025; last accessed 14/12/2023.
35. 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.
36. The key application, IVXV 1.9.10 EP2024, https://github.com/vvk-ehk/ivxv/tree/master/key; last accessed 8/7/2024, the protocols PDF: https://www.valimised.ee/sites/default/files/2024-05/RVT%20korraldus%20nr%2012%20lisa%20%28IVXV-protokollide%20kirjeldus%29.pdf.
37. 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

38. https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia; last accessed 29/2/2024.
39. Krips, K., Snetkov, N., Vakarjuk, J., Willemson, J. (2024). *"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
40. https://github.com/DrShymaa2022/articles_papers/blob/main/Letter_to_Estonia_ivoting.pdf
41. PMPhilip Daian, Tyler Kell, Ian Miers, and Ari Juels; July 2018; https://hackingdistributed.com/2018/07/02/on-chain-vote-buying/
42. 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.
43. P. Ronne, T. Finogina, and J. Herranz, "Expanding the Toolbox: Coercion and Vote Selling at Vote Casting Revisisted", E-Vote-ID 2024, Springer Verlag, DOI:10.1007/978-3-031-72244-8_9
44. https://github.com/valimised/ivxv/blob/published/auditor/src/main/java/ee/ivxv/audit/tools/IntegrityTool.java; last accessed 24/2/2025.
45. 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.
46. 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
47. Zero Knowledge Berkely MOOC 2023, lecture 5, "*KZG polynomial commitment scheme"*; https://youtu.be/tAdLHQVWlUY
48. 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
49. 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
50. 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/