Verifiable Internet Voting in Estonia

Sven Heiberg*† and Jan Willemson*‡
*Cybernetica, Ülikooli 2, Tartu, Estonia
†Smartmatic-Cybernetica Centre of Excellence for Internet Voting, Ülikooli 2, Tartu, Estonia
‡Software Technology and Applications Competence Centre, Ülikooli 2, Tartu, Estonia
Email: {sven,janwil}@cyber.ee

***Abstract*—This paper introduces an extension to the Estonian Internet voting scheme allowing the voters to check the cast-as-intended and recorded-as-cast properties of their vote by using a mobile device. The scheme was used during the 2013 Estonian local municipal elections and the 2014 European Parliament elections. 3.43% and 4.04% of all Internet votes were verified, respectively. We will present the details of the protocol, discuss the security thereof and the results of implementation.**

Keywords*—*Verifiable electronic voting

I. Introduction

The first legally binding elections allowing votes to be cast over the Internet took place in 2000 at the University of Osnabrück, Germany [1], and in Arizona, USA [2]. Just five years later, Internet voting was used in the Estonian countrywide local municipal elections [20]. Since then, legally binding Internet voting has been applied by various other countries and organizations, e.g. the Austrian Federation of Students [18], Switzerland [4], Netherlands [15], Norway [27], etc.

Several of the abovementioned implementations have encountered some security issues. For example, as a response to Arizona pilot, it was recommended to delay Internet voting until suitable criteria for security are put in place [24]. The Austrian Student Federation election of 2009 was subject to a DDoS attack [10]. Both the 2011 and 2013 attempts to introduce e-voting in Norway suffered from software and physical implementation errors [27], [8]. The 2011 Estonian elections were subject to several attacks including a proof-of-concept vote manipulation malware and politically motivated attempts to revoke the results of the whole electronic vote [13].

Electronic voting can be considered inherently more dangerous compared to conventional paper-based voting, as the lack of physical evidence creates the need to trust the electronic voting device. A buggy or malicious voting device could tamper with the electronic ballot without anybody being able to detect the manipulation. If the voting device and the digital ballot box communicate over the Internet, they are exposed to geographically unbound, highly scalable attacks from the network. A security analysis for an Internet voting system provided by SERVE (Secure Electronic Registration and Voting Experiment) suggested that Internet voting should not be attempted, unless some unforeseen security breakthrough appears [16].

Verifiable voting protocols attempt to improve the situation by providing participants with the ability to check whether certain properties hold on, e.g. the electronic tally. If the protocol gives voters the means to check the properties of their individual ballots, we can refer to an *individually verifiable* voting protocol. For example, it might be possible for the voter to check whether the electronic ballot cast over the Internet was correctly accepted by the digital ballot box. There are several protocols that provide some kind of verifiability to Internet voting [26], [5], [17], [11].

In this paper, we present an individually verifiable protocol that was used in the 2013 Estonian local municipal elections and the 2014 European Parliament elections. The paper is organized as follows. Section II describes the basic Estonian Internet voting scheme and explains the need for verifiability, and Section III defines the exact objective for the verifiability extension proposed in Section IV. Section V discusses the provided security guarantees together with the residual risk vectors, and Section VI gives practical implementation results. Finally, Section VII draws some conclusions and sets out the direction of future work.

II. Estonian Internet voting in 2005–2014

The Estonian Internet voting scheme was developed in the early 2000s and is described in detail in [13]. It has been used at seven elections during 2005–2014 and the basic protocol has remained essentially unchanged.

On the conceptual level, the scheme is very simple and mimics double envelope postal voting. The central voting system generates an RSA key pair and publishes the public part s_{pub}. The voter v authenticates herself for the voting server using her ID card or mobile ID (standard identification mechanisms widely used in Estonia), and receives the candidate list. She then makes her choice c_v (which is just a candidate number in case of Estonian elections) and encrypts it with the server's public key. For encryption, RSA-OAEP is used and a random seed r is generated for the cryptosystem. Hence the anonymous ballot ("inner envelope") is computed as $b_{anon} = Enc_{s_{pub}}(c_v, r)$. The effect of the "outer envelope" is achieved by signing the ballot using the voter's ID card, and the resulting complete ballot $b = Sig_v(b_{anon})$ is sent to the voting server (see also Figure 1).

The scheme uses re-voting as an anti-coercion measure. The voter can cast a vote over Internet several times, but only the last vote will be included in the tally. This way, if a voter feels coerced, she can re-vote later. The voter can also vote on paper to cancel her electronic vote. It is assumed that uncertainty in

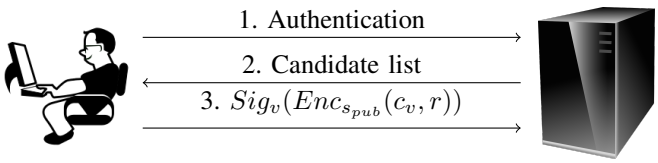

Fig. 1. The basic Estonian Internet voting protocol

the outcome of the coercion attempt makes such attempts an inefficient attack vector.

Electronic ballots are kept in the signed and encrypted form until the voting period is over. The signatures are then dropped and anonymous ballots are tallied; for that, they are decrypted with the server's private key stored in a hardware security module.

While it is rather straightforward, the system has several weaknesses, some of which were exploited during the 2011 parliamentary elections. The most severe and widely published attack was proposed by a student who made use of the fact that in its original form, the voting system gave no reliable feedback concerning whether or how the vote was actually received by the server. The student developed several versions of malware capable of blocking or even changing the vote. Due to the simple nature of the basic protocol, such manipulations would remain unnoticed by the voter [13].

After the 2011 elections, these issues were addressed in the OSCE/ODIHR report [22]. Among other suggestions, the report states:

> *The OSCE/ODIHR recommends that the NEC forms an inclusive working group to consider the use of a verifiable Internet voting scheme or an equally reliable mechanism for the voter to check whether or not his/her vote was changed by malicious software.*

The current paper can be seen as a direct consequence of this suggestion, presenting a scheme that allows the users to verify the correctness of their votes. The scheme was implemented and used as a pilot during the 2013 Estonian local municipal elections and the 2014 European Parliament elections.

However, adding vote verifiability to the system may have unexpected side effects which can violate other requirements of the election. For example, the Council of Europe has published its recommendations on legal, operational and technical standards for e-voting [3]. Recommendation number 51 reads:

> *A remote e-voting system shall not enable the voter to be in possession of a proof of the content of the vote cast.*

It can be argued that any sufficiently strong form of vote verification may be used as a proof of the content, and hence facilitate vote selling or coercion, for example [7]. In the current paper we assume the hypothesis that the truth lies somewhere in between and try to propose one possible trade-off between verifiability and coercion-resistance. See Sections V-B and V-C for a more detailed discussion.

III. Types of verifiability

There is no generally accepted definition of the verifiability of electronic voting. Various authors define it differently depending on the needs and capabilities of the community setting up the elections. We refer to [19] for a good overview and comparison of the proposed approaches. In this paper, we will rely on the definition given by Popoveniuc *et al.* [23]. They define end-to-end verifiability through the performance requirements set for the voting system. An end-to-end verifiable voting system will provide the following properties:

1) The voter is able to check that her ballot represents a vote for the candidate to whom she intended to give the vote.
2) Anyone is able to check that valid ballots do not contain over-votes or negative votes.
3) The voter can check that her ballot is recorded as she cast it.
4) Anyone is able to check that all the recorded ballots have been tallied correctly.
5) Anyone is able to check that the voters and the general public have the same view of the election records.
6) Anyone can check that any cast ballot has a corresponding voter who can perform check No. 3.

Popoveniuc *et al.* also analyze several proposed systems and conclude that some of them are fully end-to-end verifiable (e.g. Prêt à voter [25] or Scratch & vote [6]). Some other systems (e.g. Scantegrity II [9] or Helios [5]) need one of the requirements to be slightly relaxed.

We will not be requiring end-to-end verifiability in the full sense of Popoveniuc *et al.* for the Estonian voting system. We will only require the individually verifiable properties 1 (cast-as-intended) and 3 (recorded-as-cast) from the list above. There are several reasons for that. First, the 2011 parliamentary elections showed client-side weaknesses both in the preparation and transport of ballots. Cast-as-intended and recorded-as-cast properties address these weaknesses. This is similar to conventional paper-based elections that have these properties under certain assumptions, namely that:

1) The voter is capable of representing her choice correctly;
2) The ballot paper and the ballot marker pen are not tampered with and perform their function correctly;
3) The voter personally takes the ballot from the polling booth to the ballot box.

From this point on, the voter has to rely on the election officials and observers to follow the procedures correctly and to notify the public of any possible violations. The Estonian National Electoral Committee (NEC) felt that although the observability of the electronic tally can be considered in the future, the effort needed to implement end-to-end verifiability is currently not justified.

Second, achieving some additional properties would have meant implementing a completely new system with a completely new user experience compared to what the electorate is used to, and this was considered unrealistic. As we will see later in the paper, cast-as-intended and recorded-as-cast properties are achievable incrementally with respect to the current system.

IV. Verifiable Internet voting for Estonian elections

In Estonia, Internet voting makes heavy use of an existing ID card infrastructure which essentially provides one secure pre-channel between the state and the citizen in the form of certified public-private key pairs.

Since verification is something that can only happen *after* a vote is cast, we also need a post-channel that would work well together with the chosen pre-channel. During the analysis phase, a postal+SMS solution was briefly considered. It was concluded that this channel was rather expensive and still error-prone as shown by the Norwegian experience [27]. Hence another alternative was needed.

Since the basic Estonian Internet voting protocol supports vote auditing by releasing the random seed used for encryption, we decided to implement this form of verification. Of course, such a verification cannot be performed by a human alone and a computing device is required. Since verification using the same device (PC) would not address the problem of potential device corruption, we decided to introduce verification on a different platform. As of the time of the development period (2012), the prime candidates for this platform were mobile devices (smartphones, tablet computers, etc.). They provide both sufficient processing power for cryptographic operations and independent communication channels.

Verification itself requires relatively small overhead compared to the existing Estonian Internet voting system, and the entire protocol on a high level is as follows (see also Figure 2).

1) The voter authenticates herself for the server.
2) She receives a list of candidates L.
3) The voter makes her choice $c_v \in L$ and prepares the vote $b_{anon} = Enc_{s_{pub}}(c_v, r)$, encrypted with the server's public key, using randomness r. The voter sends her signed vote $b = Sig_v(b_{anon})$ to the server.
4) The server returns a unique randomly generated vote reference vr to the voter. This reference will later be used to download the correct vote to the mobile device.
5) The voter transfers r and vr from the PC to the mobile device.
6) The mobile device contacts the server over server-side authenticated HTTPS and sends vr.
7) The voter's mobile device downloads the vote b_{anon} corresponding to vr from the server together with the list of all candidates available L.
8) The mobile device computes $Enc_{s_{pub}}(c, r)$ for all $c \in L$. If for some c' the equality $Enc_{s_{pub}}(c', r) = b_{anon}$ holds, this c' is displayed to the user. If $c_v = c'$, the voter accepts the vote to have been cast as intended.

Steps 1–3 have been used since 2005 and are familiar to the general electorate. Hence, only steps 4–8 are new to voters. From the user interface point of view they can be performed rather smoothly.

The time allowed to complete steps 4–7 has been limited (30 minutes in 2013 and 60 minutes in the 2014 elections). Also, the number of times the server is ready to let the user download b_{anon} is limited (currently 3). The verifiability extension only allows for the verification of the last vote cast by the voter. Re-voting revokes both the previous ballot and the vote reference. These are largely anti-coercion measures; see Section V-B for further discussion.

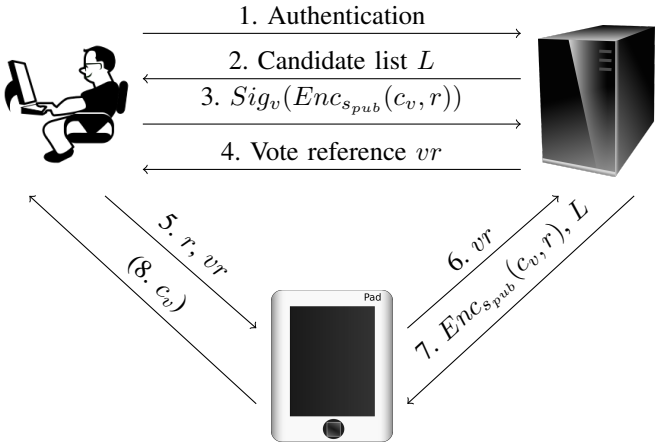

Fig. 2. The Estonian Internet voting protocol with vote verification

The most complicated one is step 5, where the random seed r and vote reference vr need to be transferred from a PC to a mobile device. Several channels can be used for that; we chose to use QR codes, since other alternatives (like a memory card, a wired connection or Bluetooth) require extra setup. When the vote is sent to the server, a QR code containing r and vr is displayed on the PC screen. The user runs a verification application on the mobile device. The application first expects to scan the QR code, which can be done by pointing the device to the PC screen. The voter does not even need to press any buttons, as the scan is completed automatically. And assuming the network connection is open, steps 6 and 7 are also automatic. Once the vote is received from the server, the mobile device follows through with step 8.

Note that the mobile device never learns the voter's identity, it just sees random values. It finds the value c' for an anonymized encrypted vote. This prevents a malicious mobile device from breaking vote privacy. Of course, it can still lie about the value of c' found, but assuming that the PC and the mobile device are not corrupt in a coordinated manner, this lie would be detected and reported by the user with high probability. The latter assumption may or may not fully hold; see Sections V and VI for more discussion and analysis in case this assumption is relaxed.

Since step 8 assumes going through the list L, it will take some time. In practice, the candidate lists in Estonia contain up to several hundred elements in extreme cases (with the values $10 \ldots 50$ being the most common). We implemented a test application computing 400 RSA2048 encryptions with the exponent 65537. On a Samsung Galaxy Ace smartphone with an 800 MHz processor this computation took roughly 1.5 seconds. Together with the time needed to communicate with the server we estimate the total running time of the verification to be up to 5 seconds which we consider a reasonable result.

It would also be possible to implement step 8 by first asking the voter to input her choice and make the comparison with one encryption, displaying a simple yes/no answer. This

seemingly more elegant solution introduces a new potential threat vector. Namely, it would be possible for a corrupt verification application not to verify anything and just say yes. In the protocol proposed above, however, in order to manipulate the vote successfully without the voter noticing, the voting and verification applications must be corrupt in a coordinated manner. We consider the complexity of such an attack prohibitively high.

In principle, it is also possible to develop vote verification software for PC platforms and carry out a public education campaign convincing voters to verify their votes on a computer different from the one that they used to cast the vote. However, we suspect that the vast majority of voters would just run the two pieces of software on the same computer, and hence the security goals set for verification would not be achieved. At the time of writing this paper, major PC and mobile platforms are running different operating systems. Thus, the voters are forced to use separate devices for voting and verification which was one of our security goals. We acknowledge that this situation may change in the future, but at least for the elections taking place in 2013–2015 this approach should be viable.

Analyzing the voting protocol, we see that the verification device does not need and should not store anything. This means that these devices can be shared among voters, making them even more accessible.

V. Discussion

In this section we will address some specific issues about the scheme and its application.

A. Failed verifications

Individual verifiability provides NEC with an additional tool to detect possible attempts to manipulate the voting result on a large scale. Verification attempts may fail due to simple user errors or hardware/software incompatibility, but failed verifications may also indicate a manipulation attack.

Most important failures in verification can manifest themselves through the following symptoms:

- Inability to download the encrypted vote from the server,
- Failure to find the corresponding candidate from the list L,
- The candidate found does not match the voter's intention.

In case of such failures, NEC suggests that voters follow a predefined set of actions:

1) Re-vote and verify using (preferably) a different PC and mobile device.
2) In case the error persists, re-cast the vote in a polling station on paper. Notify NEC of the event.

If certain errors start repeating, this information may be used by NEC to initiate research activities and take different decisions. Failures in verification do not necessarily mean that an attack is going on. E.g. a voter who would attempt to verify her vote after the vote reference vr has expired, would get a verification failure. Similarly, a voter using the wrong QR-code would get a verification failure and possibly turn to NEC for assistance.

B. Coercion-resistance

Ben Adida, author of the verifiable Internet voting system Helios, states that his system is only suitable in low-coercion settings like student governments, local clubs, online groups such as open-source software communities, and other similar situations. The protocol is not applicable for parliamentary elections. for instance [5]. The original Helios interface actually provided a "Coerce Me!" button to remind the users about the inherent threat. A similar button could be built into the Estonian voting or verification application – anyone who gets hold of the vote $b_{anon} = Enc_{s_{pub}}(c_v, r)$ and randomness r is capable of finding out the voter's actual preference.

Coercion is more likely to occur in a remote setting. Voting in polling stations takes place in the privacy of the polling booth, and the coercer has to invent ways to maintain control over the actions of the coercee. In remote environments, the coercer can observe the voter voting for a specific candidate. Estonian Internet voting uses re-voting as an anti-coercion measure.

Verifiability seems to facilitate coercion. In the Norwegian system, the coercer may ask the voter to provide the card with the verification codes and the SMS with the code actually returned. This way the coercer can be sure that the vote for the required candidate is in the digital ballot box. In the Estonian protocol, it is enough for the coercer to control the verification application.

We argue that due to the option of re-voting, coercion is not made any easier by introducing verifiability. By observing either voting or verification, the coercer cannot be sure that the vote will actually be taken into account. We also note that a coercion attack as a manipulation attack is rather inefficient. In order to achieve an additional seat in the Parliament, a great number of people have to be coerced, and thus the probability of getting caught increases. It is also time-consuming to monitor all the coercees and their actions. (Recall that both the time the server is willing to provide a particular encrypted vote for verification, and the number of times it is ready to do so, are limited.) Nevertheless, if a society sees large-scale coercion as an existing problem, any kind of remote voting – electronic or non-electronic – should be avoided at elections.

C. The threat of false verification failure claims

Of course, introducing a new component into the system also brings along new attack vectors. Merely the possibility to claim that the verification failed can be misused by malicious voters interested in, say, a reputation attack [14]. When the proposed method of vote verification was presented to Estonian politicians, this was one of the concerns they expressed. The problem is that it is very difficult to either prove or disprove such claims without violating vote secrecy. The Norwegian experience, however, showed that a widespread reputation attack based on bogus claims did not happen [27]. On the contrary, the Norwegian electorate perceived failed verifications as a positive feature – it gave feedback that had been impossible to obtain before. After having applied the verification solution in the 2013 and 2014 Estonian elections we can say that the threat of false claims did not materialize. Considering that the

verifications made during the 2013 and 2014 elections were just pilots, the incentive of potential attackers may have been lower than for legally binding runs, and thus we still need to be ready for such an attack in the future.

D. Random factor exposure

The verification scheme leaks the randomness r used in the encryption to the mobile device. Anybody in possession of r, b_{anon} and the list of candidates L can brute-force the encrypted ballot to get the candidate number. We do not see a new threat here as anybody having access to r in the voting application also could have observed the original choice encrypted together with the randomness.

E. Diverting the verification

To provide its security properties, the verification protocol relies on some assumptions. The most important assumption made is the independence of the PC and the mobile device. If an attacker was able to install malware working on both of the devices in a coordinated manner, a potential vote manipulation could go unnoticed. The report [12] claims to have developed proof-of-concept pieces of malware for both the PC and the mobile device, using the QR code channel to make hints to the verification application about the voter's choice, whereas a compromised voting client would manipulate the vote silently.

However, the report fails to describe how to achieve a coordinated installation of the developed malware on these devices. The authors of the report also admit that if this attack were to be used on a large scale, it would carry an elevated possibility of detection, since some users may attempt verification with devices owned by others. This in turn means that the goal of introducing verification has been achieved and it is still possible to have confidence in the absence of a large-scale vote manipulation attack. See Section VI for more discussions on quantified estimates on the security guarantees obtained on the example of the 2013 Estonian elections.

Another approach to attack the scheme is based on the fact that the voter is not capable of verifying if the QR presented by the voting application contains the randomness and vote reference vr corresponding to her ballot. If the malicious voting application knows the vote reference vr_1 of an already stored ballot, which encrypted the candidate number desired by the voter, then the application could encrypt any other candidate number for vr, but show the QR code with vr_1 and r_1. This way a manipulated ballot would be stored, but the verification application would show the result expected by the voter.

The limits on the number and time of verifications and the way that the re-voting is handled make this attack difficult to execute in practice. It is not possible to acquire a set of QR codes and reuse them for a longer period of time. A more robust approach would be based on the fact that most votes are never verified and it is possible to build a QR-sharing bot-net of malicious voting applications. This would make the setup of a manipulation attack more complex, and the event of using the same QR code too many times would trigger a server-side alarm.

Vote verification is not a universal measure against all possible attacks. As discussed above, re-voting is used in Estonia as an anti-coercion measure. However, this possibility can also be abused by malware installed on the voter's PC. During the original voting session, the malware may save the PIN codes of an ID card (assuming an ID card reader without a PIN pad is used, which is mostly the case). If the ID card is inserted again later (maybe for a completely different application), the malware may also use it to submit a new vote. As there is no active feedback channel currently in use in the Estonian Internet voting protocol, most voters would never know about this occurrence even if they verified their original vote. The most efficient measure against such an attack would be to implement an active feedback channel. This is one of the possible future improvements considered for the Estonian Internet voting protocol. However, since this attack is independent of verification, further discussion remains outside the scope of the current paper.

VI. Implementation results

The described verifiable Internet voting system was first implemented for the 2013 Estonian local municipal elections. For the first pilot[1], only Android OS 2.2 and higher were supported as the mobile application platform. During the elections, 136,853 electronic votes were given (including re-votes) and 133,662 counted (which comprised 21.2% of all the votes cast). Verification was utilized on 4,696 occasions (and altogether 3.43% of all the e-votes given were verified).

For the second pilot run during the 2014 European Parliament elections, support for iOS and Windows Phone was added as well. During the elections, 105,170 electronic votes were given (including re-votes) and 103,105 votes were counted (which comprised 31.3% of all the votes cast). Verification was utilized on 4,250 occasions (and altogether 4.04% of all the e-votes given were verified).

There were no failed verifications reported in 2013. This allows us to estimate the probability that a large-scale vote manipulation went undetected. Assuming that the attacker was able to manipulate k random votes, but not tamper with the verification devices and voting devices in a coordinated manner, the probability that at least one of the manipulated votes was detected is

$$1 - \left(1 - \frac{4696}{136853}\right)^k .$$

(This corresponds well to the reasoning by Neff [21].)

In order to obtain a more realistic estimate on this probability, we have to take into account possible coordinated malware (see Section V). For illustrative purposes in this paper we assume that only half of the verifications were performed on truly independent devices. The probability that at least one of

[1]According to the current Estonian legislation, verification will have legal consequences in 2015 (and the date can be moved further if necessary). The verifications during the first two elections of 2013 and 2014 were planned as pilots to try out the new technology.

the manipulated votes was detected changes to

$$1 - \left(1 - \frac{2348}{136853}\right)^k .$$

See Figure 3 which depicts both of the graphs. We can see that even if half of the devices were compromised, the manipulation of 200 or more votes would still be detected with more than a 95% probability.

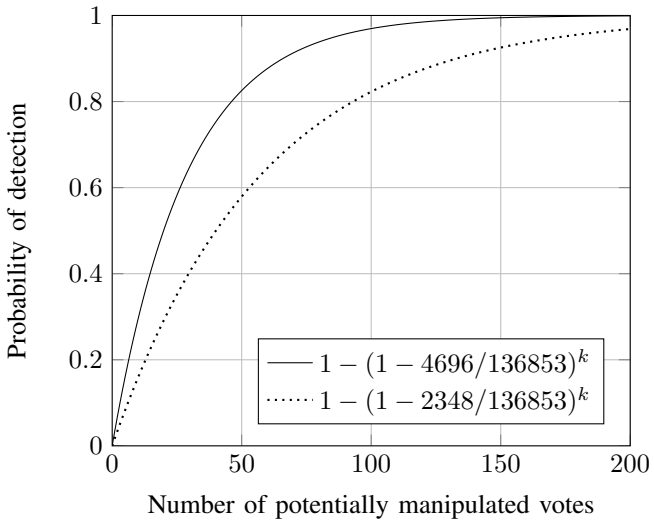

Fig. 3. Probability of large scale vote manipulation detection

The pilot in 2014 was more controversial – during the election, two software bugs were discovered in the iOS verification application. On a few occasions, the iOS application reported that it was not capable of finding the candidate number corresponding to the encrypted ballot. It appeared that binary data extracted from the QR code was interpreted as a string by the application, leading to bad encryptions under certain circumstances. The bug was fixed during the elections, the patch was successfully submitted to the iOS app store and pushed to the voters.

The second bug manifested itself when a buggy iOS verification application was accidentally used with a QR code coming from an external source (e.g. newspaper ad, online media, etc.). For the voter it looked as if her vote was not available on the server, even though it was stored correctly. This resulted in four calls to the helpdesk. The voters were instructed to cast a new vote and verify it again. No more errors were reported after this.

Hence no real vote manipulations were detected during the 2014 elections either. This allows us to estimate the probability of a large-scale attack detection exactly the same way as was done for the 2013 elections above.

VII. Conclusions and further work

In this paper, we described an extension to the Estonian Internet voting protocol, allowing users to verify that their votes are stored correctly on the server. We discussed the technical aspects and quantified the resulting security guarantees obtained during two pilot application runs.

On the one hand, Estonian democracy is rather young and all the potential weaknesses of Internet voting are aggressively used in political battles to attempt revocation or at least harm the reputation of this voting method. On the other hand, Estonian society is also very technology-oriented. For example, virtually all the eligible voters have a digital ID card capable of giving legally binding RSA signatures, and the penetration of mobile devices is growing rapidly. These considerations allowed us to propose a verifiable Internet voting scheme relying on an ID card as a pre-channel and a mobile device as a post-channel. In order to successfully and non-discoverably manipulate a vote, the attacker has to corrupt both the voter's PC and mobile device in a coordinated manner. Even if this is conceivable for a small number of votes, we consider the complexity of a corresponding successful widespread attack prohibitively high.

The system was implemented as a pilot solution for the 2013 Estonian local municipal elections and the 2014 European Parliament elections. It is expected to have legal implications in the 2015 parliamentary elections. Before legally binding conclusions can be drawn, new dispute resolution mechanisms need to be created. For example, we need to better understand how to distinguish true verification failure claims from false ones and how to deal with these false claims.

The success of the proposed system relies on the fact that currently PCs and mobile devices are independent and run different operating systems. This situation may change in the future, which means that the system will then need to be modified suitably. Also, the first pilot implementations of 2013 and 2014 are expected to give a lot of feedback, and improving the system accordingly will remain the subject of future development efforts.

Acknowledgements

This research was supported by the Estonian Research Council under Institutional Research Grant IUT27-1 and the European Regional Development Fund through the Centre of Excellence in Computer Science (EXCS) and grant project number 3.2.1201.13-0018 "Verifiable Internet Voting – Event Analysis and Social Impact".

The authors would also like to thank Arnis Paršovs for proofreading the paper and all the anonymous reviewers for their excellent comments.

References

[1] Forschungsgruppe Internetwahlen, Zweiter Zwischenbericht zum Projekt, Strategische Initiative:WahlenimInternet'nachAbschlussderWahlenzum Studierendenparlament der Universität Osnabrück am 2. Feb. 2000, 2000.

[2] Report of the National Workshop on Internet Voting: Issues and Research Agenda. Internet Policy Institute, http://verifiedvoting.org/downloads/NSFInternetVotingReport.pdf, 2001, last accessed May 6th, 2014.

[3] Legal, operational and technical standards for e-voting. http://www.coe.int/t/dgap/goodgovernance/Activities/Key-Texts/Recommendations/Rec(2004)11_Eng_Evoting_and_Expl_Memo_en.pdf, April 2005, last accessed May 6th, 2014. Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe on 30 September 2004 and explanatory memorandum.

[4] The Geneva Internet Voting System. http://www.geneve.ch/evoting/english/doc/final-livret-anglais.pdf, last accessed May 6th, 2014.

[5] Ben Adida. Helios: web-based open-audit voting. In *Proceedings of the 17th conference on Security symposium*, pages 335–348, 2008.

[6] Ben Adida and Ronald L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, WPES '06, pages 29–40, 2006.

[7] Jordi Barrat, Michel Chevallier, Ben Goldsmith, David Jandura, John Turner, and Rakesh Sharma. Internet Voting and Individual Verifiability: The Norwegian Return Codes. In Melanie Volkamer Manuel J. Kripp and Rdiger Grimm, editors, *5th International Conference on Electronic Voting 2012 (EVOTE2012)*, volume 205 of *LNI – Lecture Notes in Informatics*, pages 35–45, 2012.

[8] Christian Bull and Henrik Nore. Problems encountered. Seminar on Internet voting, http://www.regjeringen.no/pages/38377245/5_problems_encountered.pdf, September 2013, last accessed May 6th, 2014.

[9] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, and Alan T. Sherman. Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *Proceedings of the conference on Electronic voting technology*, EVT'08, 2008.

[10] Andreas Ehringfeld, Larissa Naber, Karin Kappel, Gerald Fischer, Elmar Pichl, and Thomas Grechenig. Learning from a Distributed Denial of Service Attack against a Legally Binding Electronic Election: Scenario, Operational Experience, Legal Consequences. In Kim Andersen, Enrico Francesconi, ke Grnlund, and Tom van Engers, editors, *Electronic Government and the Information Systems Perspective*, volume 6866 of *Lecture Notes in Computer Science*, pages 56–67. Springer Berlin / Heidelberg, 2011.

[11] Kristian Gjøsteen. Analysis of an internet voting protocol. Cryptology ePrint Archive, Report 2010/380, 2010. http://eprint.iacr.org/.

[12] J. Alex Halderman, Harri Hursti, Jason Kitcat, Margaret MacAlpine, Travis Finkenauer, and Drew Springall. Security Analysis of the Estonian Internet Voting System, May 2014. https://estoniaevoting.org/wp-content/uploads/2014/05/IVotingReport.pdf.

[13] Sven Heiberg, Peeter Laud, and Jan Willemson. The Application of I-voting for Estonian Parliamentary Elections of 2011. In Aggelos Kiyaias and Helger Lipmaa, editors, *VoteID 2011*, volume 7187 of *LNCS*, pages 208–223. Springer, 2011.

[14] Sven Heiberg and Jan Willemson. Modeling threats of a voting method. In Dimitrios Zissis and Dimitrios Lekkas, editors, *Design, Development, and Use of Secure Electronic Voting Systems*, pages 128–148. IGI Global, 2014.

[15] E.M.G.M. Hubbers, B.P.F. Jacobs, and W. Pieters. RIES: Internet voting in action. In *29th Annual International Computer Software and Applications Conference (COMPSAC 2005)*, pages 417–424. IEEE Computer Society, 2005.

[16] David Jefferson, Aviel D. Rubin, Barbara Simons, and David Wagner. A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), 2004, last accessed May 6th, 2014. http://www.servesecurityreport.org/paper.pdf.

[17] Rui Joaquim, Carlos Ribeiro, and Paulo Ferreira. Veryvote: A voter verifiable code voting system. In Peter Y. A. Ryan and Berry Schoenmakers, editors, *VOTE-ID*, volume 5767 of *Lecture Notes in Computer Science*, pages 106–121. Springer, 2009.

[18] Robert Krimmer, Andreas Ehringfeld, and Markus Traxl. The Use of E-Voting in the Austrian Federation of Students Elections 2009. In Rober Krimmer and Rüdiger Grimm, editors, *4th International Conference on Electronic Voting 2010*, Lecture Notes in Informatics, pages 33–44, 2010.

[19] Lucie Langer, Axel Schmidt, Melanie Volkamer, and Johannes Buchmann. Classifying Privacy and Verifiability Requirements for Electronic Voting. In *GI Jahrestagung*, pages 1837–1846, 2009.

[20] Ülle Madise and Tarvi Martens. E-voting in Estonia 2005. The first practice of country-wide binding Internet voting in the world. In Robert Krimmer, editor, *Electronic Voting 2006, Proceedings of the 2nd International Workshop*, LNI GI Series, pages 15–26, 2006.

[21] C Andrew Neff. Election confidence, 2003, last accessed May 6th, 2014. http://www.verifiedvoting.org/wp-content/uploads/downloads/20031217.neff.electionconfidence.pdf.

[22] OSCE/ODIHR. Estonia. Parliamentary Elections 6 March 2011. OSCE/ODIHR Election Assessment Mission Report. http://www.osce.org/odihr/77557, 2011, last accessed May 6th, 2014.

[23] Stefan Popoveniuc, John Kelsey, Andrew Regenscheid, and Poorvi Vora. Performance requirements for end-to-end verifiable elections. In *Proceedings of the 2010 international conference on Electronic voting technology/workshop on trustworthy elections*, EVT/WOTE'10, 2010.

[24] Caltech-MIT Voting Technology Project. Voting: What is, what could be. Technical report, Caltech/MIT, 2001.

[25] Peter Y. A. Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4):662–673, 2009.

[26] Gerhard Skagestein, Are Vegard Haug, Einar Nødtvedt, and Judith E. Y. Rossebø. How to create trust in electronic voting over an untrusted platform. In Robert Krimmer, editor, *Electronic Voting*, volume 86 of *LNI*, pages 107–116. GI, 2006.

[27] Ida Sofie Gebhardt Stenerud and Christian Bull. When Reality Comes Knocking. Norwegian Experiences with Verifiable Electronic Voting. In Manuel Kripp, Melaine Volkamer, and Rüdiger Grimm, editors, *5th International Conference on Electronic Voting 2012*, Lecture Notes in Informatics, pages 21–33, 2012.