The Application of I-voting for Estonian Parliamentary Elections of 2011

Sven Heiberg[1], Peeter Laud[1,2], and Jan Willemson[1,3]

[1] Cybernetica, Ülikooli 2, Tartu, Estonia
[2] Institute of Computer Science, University of Tartu, Liivi 2, Tartu, Estonia
[3] Software Technology and Applications Competence Center, Ülikooli 2, Tartu, Estonia
{sven,peeter,janwil}@cyber.ee

Abstract. Estonia has implemented internet voting as a method to participate in various types of elections since 2005. In Riigikogu (parliament) Elections of 2011, over 140,000 voters used the internet voting method. The share of votes cast over the internet among all votes was 24.3%. In light of this popularity it is questioned by various stakeholders whether internet voting can be implemented correctly and securely to support electoral principles such as uniformity. This paper gives an overview of the Estonian Internet Voting System and analyzes events that occurred during the Riigikogu Elections of 2011.

1 Introduction

There are four main voting methods implemented by the Estonian National Electoral Committee (NEC): voting on election day, voting abroad, advance voting and internet voting (i-voting). Most of those methods are paper-based, subject to certain organizational and procedural regulations. Internet voting is exceptional as it offers a possibility for a voter to participate in election digitally from his own computer over the internet anytime during the period of advance voting.

As of August 2011, i-voting has been used in five Estonian elections starting from the Local Government Council Elections in October 2005 and currently ending with the Riigikogu Elections in March 2011. Before the first use, a legally non-binding pilot was conducted in January 2005 in Tallinn. In October 2005, there were 9,317 voters who i-voted (i-voters) and 9,287 counted ballots cast over the internet (i-votes). 1.9% of participating voters and 7.2% of advance voters were i-voters. Since then, the number of i-voters has increased steadily. In March 2011, there were 140,846 i-voters and 140,764 counted i-votes. 24.3% of participating voters and 56.4% of advance voters were i-voters. Those numbers indicate that i-voting is accepted by the electorate. It is also evident that the tally of i-voting has the potential to significantly influence the election outcome.

During the Riigikogu Elections of 2011, several i-voting related incidents took place. Appeals were filed to NEC demanding that the i-voting results should be revoked because of alleged vulnerabilities of the applied scheme and the legislative problems [3, 2, 4]. In one case, the revocation of election results altogether was demanded [14]. All those appeals were dismissed by the Supreme Court [21, 22, 20].

The history of Estonian i-voting goes back to 2001, when two reports concerning i-voting were published. The analysis ordered by the Ministry of Justice [16] stated that it was unrealistic to implement statewide i-voting in 2002 because of the lack of suitable technology; instead, a research program towards i-voting was suggested. The analysis ordered by the Ministry of Transport and Communications [23] suggested that it was possible to implement statewide i-voting in 2002. Both reports agreed that i-voting is inevitable in the future and development thereof should be considered as a long term process.

In 2002, i-voting was regulated in the Riigikogu Election Act with the condition that the method shall not be applied before 2005. In 2003, a group of experts proposed an i-voting scheme for Estonia [7]. The security analysis of the scheme stated that in order to implement i-voting, it was necessary to find optimum between the theoretical security of the voting scheme and the complexity of its implementation. On one side of the compromise, there were comprehensibility, similarity to conventional voting, maximal use of the digital signature solutions available in Estonia, simplicity of the cryptographic protocol, and feasibility to implement the system with the know-how present in Estonia itself [6].

The analysis also stated: “The other side of the compromise or, in principle, the weak point of the scheme, is the need to trust central servers and computers of the voters. Is such a compromise reasonable? In our opinion – yes.” [6]

A few months later, in 2004, another group of experts published a report which analysed computer and communication security issues in the internet-based voting system SERVE (Secure Electronic Registration and Voting Experiment) built for the U.S. Department of Defense. The report stated that internet- and computer-based voting

systems have numerous fundamental security problems, which open the possibility for large scale attacks such as selective voter disenfranchisement, privacy violation, vote buying and selling and vote switching [13].

The authors of the SERVE-report made a new public statement in 2007 where they found another Department of Defense report on electronic voting technology quite troubling and assured that the arguments presented in 2003 still hold and there is no way to secure internet voting [12].

In 2010, a modified version of security analysis for the Estonian i-voting scheme found that the challenge of successful i-voting has been solved and implemented in practice [5].

OSCE/ODIHR Election Assessment Mission observed the Estonian 2011 parliamentary election with i-voting as one focus. The final report of the mission gave altogether 13 recommendations regarding to the legal framework, oversight and accountability, and some technical aspects of the internet voting system [19].

In subsequent sections, we will give an overview of the i-voting scheme implemented in Estonia, explain its architecture and approach to security, give an overview of the most important incidents during Riigikogu Elections of 2011, and propose a road ahead based on the analysis of these incidents.

2 Estonian I-voting Scheme

2.1 Legal and Technical Framework

Requirements for the design and implementation of a voting method come from the legislation which defines the election and the electoral system. The Constitution of the Republic of Estonian states the basic electoral principles – freedom, generality, uniformity, directness and voting by secret ballot. The Riigikogu Election Act defines Riigikogu Elections in detail by regulating candidates, registration, voting procedures, etc. Similar legislation exists for other types of elections.

Since 2002, Estonian citizens are issued National Identity Cards (ID-cards) carrying a chip capable of performing RSA operations. Each chip contains two RSA secret keys, used respectively for authentication and for signing. The corresponding public keys are bound to the cardowner by certificates issued by the Estonian National Certification Authority (NCA). Various government and private sector services use the ID-cards for identification and legally binding digital signatures. Currently, the ID-card is the primary identity document for both computer mediated and direct communication which makes it available to all voters. The principle of using ID-cards to do i-voting has been present in Estonian legislation since 2002.

To fight bribery and coercion, the concept of i-vote revocation is legislated. A voter can cast an i-vote several times, only the last one will be counted. I-vote is also revoked if the voter uses any paper-based voting method (casts a p-vote) during the advance voting period. Those measures were not accepted unanimously by the Riigikogu; there were doubts whether it violates the uniformity of elections and secrecy of ballots [18]. The objection was that paper-based voting methods do not allow revocation. As a mild compromise, the law limited the possibility for i-vote revocation to the advance voting period.

The Penal Code has defined several election-related criminal offences such as interference with election. NEC has authority to declare the voting invalid on polling division, electoral district, county or state level if some detected violation of the law significantly affected or may significantly affect the voting results. In this case, repeat voting is held. I-voting results can currently be declared invalid only as a whole. If the i-voting is cancelled due to some violation before the actual election day, then the electorate is notified and voters can revote. In this case, no repeat voting is held.

2.2 Architecture of the I-voting Scheme

We shall describe the architecture of Estonian i-voting scheme only as much as it is necessary to understand the case studies in the subsequent sections. Readers can refer to [7] for a more complete description.

The core components of the architecture of the i-voting scheme are i-voting protocol, i-voting system and i-voting client application (IVCA) - an election specific application which allows voters to cast their votes from Windows, Linux and MacOS X based computers.

I-voting system (IVS) is responsible for i-vote collection, storage and tabulation. The system is interfaced with NCA, Election Management System, Population Register and NEC website. The roles of the IVS are fulfilled by three different servers:

- The Vote Forwarding Server (VFS) is responsible for authenticating i-voters, distributing candidates' lists to i-voters and accepting the i-votes; VFS is available over the internet.

– The Vote Storing Server (VSS) is responsible for storing the i-votes over the period of time and for the anonymization of the i-votes before the actual tabulation; VSS is kept behind a firewall, connections from VFS are allowed.
– The Vote Counting Server (VCS) is responsible for the tabulation process. VCS is offline at all times.

IVS has a RSA keypair to protect ballot secrecy. The public key (ivs_{pub}) is published with IVCA. The private key is stored in tamper-resistant hardware security module (HSM) used only by VCS and protected by multiparty authentication scheme. In practice, 4 of 7 NEC members have to be present to activate the private key.

For Riigikogu Elections, each voter belongs to one of 12 electoral districts. Each candidate has a unique candidate number and is registered to one electoral district. Only voters from the same district can vote for the candidate. We refer to the list of candidates that voter v can vote for as C_v.

Setup Phase. I-voting takes place in four phases: setup, voting, revocation, and tabulation. In the setup phase, the IVS is prepared for election. The VSS is set up with the list of voters and an empty digital ballot box. The VCS is set up with the list of candidates. The IVS keypair is generated in HSM. The VFS is set up with the list of voters, the list of candidates, and HTTPS authentication certificate. The IVCA is set up with election specific data including ivs_{pub} and VFS certificate. The IVCA is digitally signed by NEC; fingerprints and download location are published in newspapers and on the NEC website.

Voting Phase. In the voting phase, the i-voting protocol is executed between the IVCA and the VFS. Mutually authenticated HTTPS is used as the transport protocol. The VFS verifies that v is an eligible voter and returns the candidate list C_v with an indication whether it is a repeated vote or not. After the voter v has selected a candidate $c \in C_v$, encryption is used to produce an anonymous ballot: $b_{anon} = RSA_{enc}(ivs_{pub}, c)$. The anonymous ballot is signed with the voter's ID-card. The i-vote consisting of b_{anon}, a digital signature and a signing certificate of v is sent to the VFS which verifies the signature and forwards the ballot to the VSS for storage. The VSS verifies the signature and checks the status of the signing certificate by NCA. If no problems occur, the i-vote is stored before revoking any possible previous i-votes cast by the voter. HTTP is used as the transport protocol between the VFS and the VSS.

At the end of the voting phase, the list of all i-voters is generated and sent to the polling stations for reference. The VFS and the VSS are disconnected from the network.

Revocation Phase. The voting phase is followed by the revocation phase during which i-votes of those who also have p-voted are revoked.

Tabulation Phase. At the end of the revocation phase, the contents of the ballot box are anonymized – digital signatures are separated from encrypted votes so that the VCS will not be able to see which voter voted for which candidate. Anonymized ballots are stored in 12 distinct lists according to the election districts that the original voters belonged to. Those lists are burned to a DVD and carried to the VCS. For the tabulation itself, the IVS's private key is activated and the anonymous ballots are decrypted. After the decryption, valid candidates for the district are tabulated.

2.3 Security Considerations

The Estonian i-voting scheme relies on the assumption that we can trust the owner of the IVS and we can trust voter's computer. However, some measures to reduce the necessary trust have been taken.

Trust in the IVS. For electoral principles to hold, the IVS has to function correctly: accept all of the votes cast by eligible voters, preserve the integrity of the ballot box at all times, anonymize the votes before the tabulation, correctly execute the correct tabulation algorithm on the correct input and publish the produced output. To achieve this correct behaviour, a set of organizational regulations and procedures are established, all of which are audited. For example, the anonymization of i-votes can only occur in the presence of at least 2 election officials, an auditor and possible external observers. All procedures are defined beforehand in written form, and all actions and outcomes are recorded on tape. Without enforcing those regulations, the IVS owner could manipulate the election results on a large scale by adding or removing votes from the digital ballot box without getting caught.

To support organizational protocols responsible for ballot box integrity, five audit logs containing SHA1-checksums of i-votes calculated over b_{anon} are stored in the VSS and the VCS. The contents of the log-files are following:

- L_1: checksums of all i-votes accepted by VSS;
- L_2: checksums of all i-votes revoked by VSS;
- L_3: checksums of all i-votes sent to tabulation by VSS;
- L_4: checksums of all i-votes declared invalid by VCS;
- L_5: checksums of all i-votes declared valid by VCS.

L_1, L_2 and L_3 also contain the personal code of the voter. This means that a person who has access to all of the audit logs is able to link the original voter to the hashvalue of encrypted ballot b_{anon}. At the end of the tabulation phase, the following conditions must hold:

$$L_1 = L_2 \cup L_3 \text{ and } L_2 = L_4 \cup L_5 \ .$$

If those conditions do not hold then it can be said that the contents of the digital ballot box have been tampered with.

Trust in Voter's Computer. The Estonian i-voter has to trust the computer used for i-voting with the IVCA. Malicious software executed in the computer could manipulate the IVCA to break secrecy and integrity of the ballot. If the malicious software could be distributed widely, the attacker would have the potential to manipulate election results on a large scale. This could occur in several ways, e.g. by sending modified votes to get his candidate elected or by sending encrypted garbage to discredit the i-voting altogether.

Detection-based Security. The state of the art in malware distribution leaves no doubt that the environment where the IVCA is executed cannot generally be considered safe. Malware can use several attack-vectors to alter the behaviour of the IVCA with no feedback to the voter. One possible vector would be using debugging interfaces offered by the underlying operating system. Debugging a process means stepping through its instructions one at a time while examining the contents of the memory. It is common that while debugging a process, a developer decides to overwrite some memory locations with new values, thus altering the actual state of the process. A similar approach can be taken by malware attacking the IVCA: if a voter has selected the candidate c_1 then change the selection to c_2 right before the encryption. The success of the attack depends on the capability of finding the right breakpoint to stop the process and finding the right memory location to alter. With the lack of the IVCA source code, an attacker can approach those problems by reverse engineering the executable file of the IVCA.

It is possible for a process to detect whether it is being debugged. An attacker who by reverse engineering discovers that some detection is used can take countermeasures in his malicious code.

To reduce the risk that a vote cast by the IVCA is tampered with by malicious code to an acceptable level, the following actions have been taken: (i) a detection system for known attack-vectors is built into IVCA; (ii) methods are used to complicate reverse engineering. During the voting phase, the IVCA instances report their opinion on the hostility of the environment to the IVS. The NEC, CERT-EE and volunteers from the Estonian Cyber Defence League also monitor the IVS and Estonian internet for known malware activity. This information is input to the NEC to decide whether i-voting is under attack or not.

It is hoped that a 7-day i-voting period is short enough to avoid reverse engineering of the IVCA, designing and implementing robust and stealthy malicious code, distributing and activating it on a large scale.

3 Riigikogu Elections of 2011

In this section, we study two cases which occurred during the Riigikogu Elections of 2011. On February 2nd, a student turned to NEC claiming that he had written an election rigging malware which was able to tamper with the IVCA. On March 6th, during the tabulation phase, one of the i-votes was declared invalid. Those two cases are not directly related to one another but indicate possible problems with the current i-voting scheme and therefore deserve some analysis.

3.1 Case: Invalid I-vote

Invalid I-vote is Found. One of the i-votes was registered invalid by the VCS during the tabulation phase of the Riigikogu Elections on March 6th, 2011. In the case of p-voting, an invalid vote is nothing special. It is quite common that voters cast invalid votes intentionally to express the attitude towards the ongoing election by leaving the ballot paper empty or writing different free-text statements on it.

The Estonian electoral system does not give any meaning to invalid votes; they are not considered as part of the election result. The IVCA has no functionality for casting an empty or otherwise invalid ballot. A voter who wants to intentionally cast an invalid i-vote, must write a new IVCA that makes it possible to encrypt random data, or find a way to manipulate the current IVCA to cast an invalid vote.

Analysis of the Cause. The possibility that an i-vote could be invalid was foreseen in the i-voting protocol. Also, the software was developed to distinguish valid i-votes from invalid ones. On the other hand, this was the first time when one of the i-votes was found invalid, so a bug in the software or the procedures for handling the IVS during the election was suspected.

The analysis of the VCS error logs showed that the invalid i-vote appeared to be correctly encrypted with ivs_{pub}. This left two conceptual possibilities for the vote to become invalid: (i) the plaintext did not follow the formatting rules for i-vote; (ii) the plaintext followed the formatting rules, but pointed to a non-existent candidate number. Further analysis pointed out five possibilities for the invalid vote to occur:

- a bug in the IVCA – sending a malformed ballot to encryption;
- a bug in the VFS – sending an invalid candidate list to the IVCA;
- a bug in the VCS – misinterpreting the decrypted vote;
- human mistake – the VCS and the VFS were set up with incompatible candidate lists;
- someone intentionally cast an invalid i-vote.

I-vote decryption seemed to be necessary to rule out most possibilities. It would be relatively safe to say that there is a bug in the VCS if the plaintext was a valid vote pointing to a valid candidate and the candidate lists in the VFS and the VCS were compatible. Only human mistake in the IVS setup procedures could be excluded without decrypting the i-vote. On the grounds of this analysis, the NEC decided on April 1st to decrypt the invalid i-vote and examine its contents. The decision was later reverted due to the possible threat to electoral principles.

Ballot Secrecy. The Constitution of the Republic of Estonia holds voting by secret ballot as one of the main electoral principles. To achieve this requirement, RSA encryption is used. To achieve another principle – uniformity – b_{anon} is stored together with the digital signature which unanimously identifies the original voter. To maintain ballot secrecy, i-votes are anonymized before the tabulation.

The fact that the IVS auditing logfiles can link original voters to the hashes of the encrypted ballots means that if the invalid i-vote was indeed decrypted separately from other i-votes, the ballot secrecy would be protected only by organizational means.

Invalid I-vote as a Possible Attack. In parallel to setting up the analysis framework, additional tests and preliminary code reviews to the IVS were conducted. No bugs were identified and the possibility of an intentionally spoiled vote was taken into consideration. It occurred that there is at least one relatively easy way to influence the contents of b_{anon} without writing a new application or directly attacking the existing one.

If the ID-card is used for i-voting, then the HTTPS connection between the VFS and the IVCA is mutually authenticated. In the case of cell phone based digital identity – Mobile-ID – only the VFS is authenticated; the voter identification follows from the Mobile-ID protocol and cannot be used on the HTTPS level. On the Windows platform the trust to the VFS is configured via the system certificate stores – if the HTTPS certificate of the VFS is signed by some trusted certification authority (CA), then the connection is trusted. This opens the possibility for a man-in-the middle attack where the user's certificate store is compromised with the attacker's CA certificate and an intercepting HTTPS proxy using a certificate signed by attacker's CA is installed between the IVCA and the IVS. The proxy modifies the original candidate list sent to the IVCA so that it contains invalid candidate numbers. This is possible because the candidate list sent from the VFS is not digitally signed as the HTTPS channel security is considered sufficient to guarantee the integrity of the message. The user does not notice the invalidity of the candidate numbers and casts a vote which is correctly formatted and encrypted by the IVCA and forwarded to the

VFS by proxy. This type of intentionally invalidated i-vote would have been falsely identified as a bug in the IVS after the i-vote decryption.

If the attacker wasn't aiming for the discrimination of the voter but for publicity, then the previous scenario would be used by the attacker himself to decoy the election officials to show whether the NEC can find out who did cast the vote from the contents of the ballot. If some more sophisticated technique to invalidate the ballot would have been applied, then the contents of the ballot could have been anything from the personal identification of the attacker or personal identification of someone not involved at all to a well formed ballot with an invalid candidate number.

Reverting the Decision. After considering the matter of ballot secrecy and the possibility of an attack against i-voting as such, the NEC reached the conclusion that it would be better not to create a precedent of decrypting one i-vote separately from others. The decision from April 1st was reverted on April 8th. It was decided to carry through only those analyses that do not require the ballot decryption. After the analysis, it was clear that there were two possibilities: (i) a hard to find bug (such as memory corruption) in the IVCA, or (ii) an intentionally spoiled i-vote.

3.2 Case: Student Writes a Ballot-manipulating Script

Revocation of I-voting Results Demanded. On February 26th, student P. sent an e-mail to the NEC and three major newspapers claiming that he had written a prototype of an election rigging malware. Attached to the e-mail there was a presentation which referred to the SERVE report [13]. The presentation pointed out that a malicious piece of software controlling both input and output interfaces on a client computer was a threat to the IVCA as it was capable of manipulating the voter to believe that he has voted for candidate c_1, although the malware actually voted for candidate c_2.

Election officials analyzed logs of both ongoing election and the test-election from February 8th to 10th. It appeared that the most remarkable voting session from the test-election belonged to P. The session lasted for 14 hours and indicated several attempts to tamper with the IVCA. Also, every time P. had voted during the real election, his voting sessions were marked suspicious.

On February 28th, P. gave election officials access to the source code of his malware. P. also demonstrated the attack to journalists. Election officials claimed that the type of malicious behaviour implemented by the malware was detected by the IVS. In response to these claims P. implemented a new type of attack – the malware now selectively held back ballots for certain candidates, whereas the voter was left with the impression that his vote was successfully sent to the VFS. This attack was demonstrated to observers of OSCE/OHDIR on March 1st, and it was screened on National Television on March 9th after the election had ended.

On March 5th, P. filed an appeal to the NEC [3] demanding the revocation of all i-votes, claiming the following:

- Ballot secrecy was not guaranteed as the IVCA can be a subject to screen monitoring software;
- The IVS contained no protection against voter disenfranchisement type of attacks;
- CERT-EE's capability of monitoring malware distribution in the Estonian internet was not proven;
- The voter cannot check whether his vote was accepted by the IVS, hence the system did not comply with Riigikogu Election Act.

On March 6th, preliminary election results were published. On the same day, the NEC decided to reclassify the appeal as a note as it did not address any violations of personal rights. A reply was sent to P. clarifying that the NEC is aware of potential attack objects, methods and time, which makes the task of detection significantly easier and no attempts to attack the IVS have occurred in any election. The NEC assured its awareness of alternative protocols reducing the need to trust the voter's computer and did not exclude their use in the future. The NEC stated that the IVS has been implemented to be trustworthy and compliant to Riigikogu Election Act [8].

P. modified his original appeal and refiled it to the NEC on March 8th, still demanding the revocation of all i-votes [2]. The NEC, according to its procedures, forwarded the appeal to the Supreme Court which on March 21st dismissed the appeal arguing that although P., as an adult Estonian citizen, could be subject and therefore an interested party to voter disenfranchisement attacks, himself knowingly executed this type of malware in his computer, therefore his rights were not violated. It is necessary to detect the violation of the person's right to vote in order to to revoke the election results, hypothetical possibility alone is not sufficient for the revocation [21].

The actions of P. did not go unnoticed. On March 24th, one of the parliament-parties filed an appeal and demanded the revocation of the election results as a whole [14]. The appeal referred to findings of P. and was dismissed by the Supreme Court on March 31st [20].

Technical Details of the Attack. The initial version of the malware attacked both the ballot secrecy and integrity; the version presented in the television attacked ballot secrecy and implemented voter disenfranchisement based on the candidate selection. P. gave the NEC access to the initial version of the malware, therefore, this version is described here. The modifications from the initial version to the version presented in television should be relatively straightforward to anybody with some programming experience.

In its general setup, the attack was related to the attack of Estehghari and Desmedt against Helios internet voting system [9]. Both of the approaches modify the appearance of the voting software on screen. However, since Helios is a web-based application, Estehghari and Desmedt were able to mangle with the document object model of the webpage to achieve the desired result. The proof of concept malware of P. used the IVCA graphical user interface (GUI) as an attack-vector. It was written in AutoIt scripting language [1] which is a framework for scripting GUI-based Windows applications. Optical character recognition (OCR) technology was used on the IVCA screenshots to decode voter personal data and the intended candidate. Fake-IVCA was built from the screenshots and the message-loop of the original IVCA was poisoned with generated mouse events. The fake-IVCA was used to leave the voter with the impression that the ballot was cast as intended. Underneath the fake-IVCA, the original software accepted generated events and voted for a semi-randomly selected candidate.

The malware took advantage of the fact that the IVCA is a wizard-like application with a very simple state machine. Although it is possible to move back and forth between various views, most people cast their ballot in one go. The malware was implemented as a series of stages, whereas each stage and corresponding actions were associated with a certain state of the IVCA. The success of the malware depended on its ability to detect state changes in the IVCA and hide its own existence from the voter.

To detect state changes, certain IVCA GUI regions were examined with AutoIt `PixelChecksum` function. If a voter reached the candidate selection, the ballot manipulating code was activated. The fake-IVCA – a screenshot of the candidate list and the selected candidate – was hiddenly generated. Regions of the original IVCA were examined to determine whether the voter had selected another candidate and the fake-IVCA was updated respectively. When the voter pressed a button to encrypt and sign the ballot, the fake-IVCA was made visible on top of the real IVCA. Mouse events were generated to the fake-IVCA to select any other candidate. This was possible as the fake-IVCA consisted of a non-transparent clickthrough foreground window which accepted mouse events but passed them without modification to the window underneath it. This way, the main window of the original IVCA received the events, but the foreground indicated no changes in the application state. After the encrypted and signed ballot had been successfully sent to the VFS, the original data was analyzed with OCR and saved to a log file.

Unlike the attack by Estehghari and Desmedt [9], the malware of P. did not contain a distribution mechanism and it did not hide its existence nor behaviour any more than it was necessary for a prototype. For example, the GUI checksum method was not robust enough to be applied over a large set of computers with different screen resolutions, operating system versions, display drivers and installed fonts; fake-IVCA creation and OCR methods used the file system to store the data, no steps were taken to hide those files, and the mouse event insertion method to modify the ballot was not robust enough to vote for a specific candidate. Although these shortcomings rendered this specific piece of malware useless for a real large scale attack, it is possible to overcome them with a reasonable amount of extra effort.

4 Discussion

Besides the two major issues discussed in Section 3, there were others as well. Three people turned to the i-voting help-desk with the following problem: the IVCA GUI was too large to fit onto their computer screen and two candidates on the bottom of the list were hidden by the Windows task-bar. The problem was caused by fixing the minimal supported resolution for the GUI design. This event was picked up by one of the candidates who demanded nullification of i-voting results [4].

The public reaction to this shortcoming in the IVCA shows clearly that for i-voting it does not suffice to be clean; it also has to look clean. Any shortcomings in quality of the system and transparency of the processes have a potential to become a weapon in political battles. The abovedescribed shortcoming in the IVCA GUI is easily avoidable in the next version of the IVCA, whereas the two cases presented in Section 3 identify conceptual problems in the Estonian i-voting scheme.

Invalid I-vote. By now, it is clear that the root cause of the invalid i-vote will never be exactly known. Although several initial versions were excluded in the process, the distinction between a bug in the IVCA or intentionally

invalidated i-vote could not be made. Minor technical corrections to the i-voting protocol and the IVS do not change the fact that the ballot secrecy relies heavily on organizational procedures in the Estonian i-voting scheme. It is theoretically possible for the NEC not to anonymize i-votes and use a modified VCS to break the secrecy of all ballots. To break the secrecy of one ballot, it is sufficient to decrypt it separately from others and later analyze audit log-files.

The case of an invalid i-vote could have been avoided with an i-voting protocol using zero-knowledge ballot validity proofs [15]. The validity of i-votes could have been verified, while the identity of the voter was still known, leaving more room for the action to NEC. On the other hand, this raises new questions such as the legality of validating the contents of the ballot before storing it in the digital ballot box. An IVS implementing a mix network-based anonymization system [15] would have reduced the required trust in the NEC and allowed the analysis of invalid i-vote in a secure manner with respect to ballot secrecy.

Student's Attack. P. exploited the fact that the Estonian i-voting scheme contains no hard countermeasures against malicious computer. The anomaly detection system only makes it possible to indicate that something is happening, and it is not possible to exclude the possibility of something happening. Any detection-based protection scheme has the following weaknesses:

- an attack with no signature in the detection engine is undetected;
- the efficiency of the scheme relies on the actual response to the detected incidents.

As P. used a previously known attack-vector, his tampering with ballots was detected by the IVS, but there were no prompt mechanisms to discover the voter disenfranchisement attack. Neither was the IVS actively monitoring the percentage of i-voters not finishing their transactions, nor offering a proof to a voter that the ballot cast was accepted as intended. The lack of countermeasures against voter disenfranchisement allowed P. to execute his attack successfully.

Most sessions associated with P. were marked as suspicious by the detection engine, but no action besides observation was taken by the NEC until the attacker wrote an e-mail himself. Then the communication with the attacker became possible and the NEC got to analyze the reasons for the alerts in the log-files.

Currently, the only real action that the NEC can take in case of large scale attacks against the IVS, is to revoke i-voting results as a whole and call people to p-voting. A single anomaly has no significant impact to the election results and is therefore not acted upon. In light of more than 140,000 i-voters from more than 100 countries this lack of repertoire is dangerous.

Toward Secure I-voting. As with the IVS server side problems, there exist i-voting protocols which handle the problem of trusting voter's computer.

The IVCA is executed in the malicious environment, where malware could manipulate its behaviour. One possible solution is to use a blind voting scheme such as one proposed by Okamoto [17] where for each voter personalized candidate numbers (codes) are generated. Codes can later be re-unified by the tabulation process for tally. The voter gets his codes through some pre-channel and uses computer to enter and send the code for the desired candidate. It is impossible for a malware to know all the codes which leaves denial of service as the main attack-vector. The problems with this protocol are (i) one cannot i-vote without the codes; (ii) code generation must be performed in a privacy preserving manner; (iii) most of the people will not find this system usable.

Tamper-indicating voting schemes such as [10] and [11] take a weaker approach to security. Each voter again gets personalized candidate numbers (codes) over a pre-channel. Each voter also has a post-channel with the IVS such as SMS. After voting with a point-and-click GUI, the IVS sends a receipt to the voter over the post-channel. The receipt can then be compared to the voter's codes to see if it matches the candidate the voter intended to vote for. In this scheme, a malware can still manipulate the ballot, but will be detected when doing so. This is enough to detect manipulations on a large scale which is crucial in assuring free election.

From the viewpoint of ballot secrecy, tamper-indicating voting schemes run into conflict where the IVS sends the voter a code corresponding to the candidate the voter voted for, but the NEC claims that the voter's identity and the contents of the ballot cannot be connected. Cryptographic protocols achieve this property of 'knowing without knowing' for example with zero-knowledge proofs or homomorphic encryption. If these methods are not understandable to the general public, this property can be used as a weapon in political battles.

The multi-channel nature of tamper-indication needs further analysis to clarify the security requirements and explain the risks if the requirements are not met. For example, if it would be possible to control both the code generation and the post-channel, a large scale attack against ballot secrecy would be possible.

Multi-channel protocols also have a higher computational cost which must be analyzed in light of real life requirements. The protocol of [11] does not scale for the Estonian case of ≈800 candidates and ≈900 000 voters. Gjøsteen [10] notes that protocols based on homomorphic tallies are not efficient for Norwegian elections. Organizational complexity which requires independent parties to host separate components of the system makes it difficult for a single governing body to organize elections.

Tamper indication is another method of detection where the question of a legal framework for incident response is as important as with the current scheme. The possible actions that the NEC can take, in case an anomaly is detected, have to be regulated.

The protocol [11] is also subject to false-positives which introduces a new attack-vector. Consider the voter claiming that he has voted for candidate a and has in fact received the code for candidate b. The SMS contains information for detection, but detection itself is not a proof. If we can only detect the manipulation and not prove it, this opens new ways for manipulation. On the other hand, if a voter is able to prove that his vote was accepted as intended, bribery becomes possible which is dangerous in complex social situations.

Avoiding bribery and at the same time avoiding malware is an example of contradictory requirements that an i-voting system must fulfil. If there is no method to completely satisfy both requirements, a political decision has to be reached about which risk is acceptable in the given context. For these decisions, the i-voting requirements must be considered as a whole. The treatment of one vulnerability in isolation from other requirements will not result in a functional i-voting system.

5 Conclusions

Estonia has put a lot of effort into developing a usable and efficient i-voting system. The i-voter turnout in Riigikogu Elections of 2011 shows that the Estonian electorate has accepted i-voting as a voting method. In this article, we described the Estonian i-voting scheme and discussed how it complies with the electoral principles – a fair amount of the i-voting architecture is concerned with meeting the security requirements deduced from those principles.

We saw that during Riigikogu Elections of 2011, several weaknesses present in Estonian i-voting scheme were materialized. The analyzed events indicate real-life attacks that an i-voting system has to withstand. From these events we conclude that it is necessary to work toward new, more secure i-voting protocol. We need to reduce the level of trust required in the voter's computer and provide the NEC with means to show that it could not act malicious even if it wanted to. It is possible that i-voting related legislation may be refined to meet these requirements.

The goal of secure i-voting cannot be reached by dealing with single vulnerabilities in isolation. Requirements for an i-voting scheme must be handled as a system. Due to the interdisciplinary and possibly contradictory nature of the system, both the creation of the system and the design of the technical solution according to the system require a widely-accepted political decision. It is also important to explain the system and the possible choices among all options to the general public in order to reduce the risk of manipulation with the election results.

References

1. AutoIt Automation and Scripting Language. http://www.autoitscript.com/site/autoit/.
2. Appeal no. 14-11/406-2 to NEC, March 8 2011. In Estonian, http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad.
3. Appeal no. 14-11/406 to NEC, March 5 2011. In Estonian, http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad.
4. Appeal no. 14-11/446 to NEC, March 10 2011. In Estonian, http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad.
5. Arne Ansper, Ahto Buldas, Aivo Jürgenson, Mart Oruaas, Jaan Priisalu, Kaido Raiend, Anto Veldre, Jan Willemson, and Kaur Virunurm. E-voting concept security: analysis and measures, 2010. Estonian National Electoral Commitee, EH-02-02.
6. Arne Ansper, Ahto Buldas, Mart Oruaas, Jaan Priisalu, Anto Veldre, Jan Willemson, and Kaur Virunurm. E-voting concept security: analysis and measures, 2003. Estonian National Electoral Commitee, EH-02-01.
7. Estonian National Electoral Committee. E-Voting System. General Overview, 2010.
8. Estonian National Electoral Committee. Answer to note 14-11/406, March 7 2011. In Estonian, http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad.

9. Saghar Estehghari and Yvo Desmedt. Exploiting the Client Vulnerabilities in Internet E-voting Systems: Hacking Helios 2.0 as an Example. In *2010 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '10)*, 2010.
10. Kristian Gjøsteen. Analysis of an internet voting protocol. Cryptology ePrint Archive, Report 2010/380, 2010. `http://eprint.iacr.org/`.
11. Sven Heiberg, Helger Lipmaa, and Filip van Laenen. On E-Vote Integrity in the Case of Malicious Voter Computers. In *ESORICS'10*, pages 373–388, 2010.
12. David Jefferson, Aviel Rubin D., and Barbara Simons. A comment on the May 2007 DoD report on Voting Technologies for UOCAVA Citizens, 2007. `http://www.servesecurityreport.org/SERVE_Jr_v5.3.pdf`, Last accessed on August 27th, 2011.
13. David Jefferson, Aviel Rubin D., Barbara Simons, and David Wagner. A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), 2004. `http://www.servesecurityreport.org/paper.pdf`, Last accessed on August 27th, 2011.
14. MTÜ Eesti Keskerakond. Appeal 14-12/535, March 24 2011. In Estonian, `http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad`.
15. Helger Lipmaa. *The Handbook of Information Security*, chapter Secure electronic voting protocols. John Wiley & Sons, 2006.
16. Helger Lipmaa and Oleg Mürk. E-valimiste realiseerimisvõimaluste analüüs (An analysis of the possibility to organise e-voting), 2001. Analysis ordered by Estonian Ministry of Justice. In Estonian.
17. Tatsuaki Okamoto. Receipt-Free Electronic Voting Schemes for Large Scale Elections. In *Security Protocols Workshop'97*, pages 25–35, 1997.
18. OSCE/ODIHR. Republic of Estonia. Parliamentary Elections 4 March 2007. OSCE/ODIHR Election Assessment Mission Report, 2007. ODIHR. GAL/56/07.
19. OSCE/ODIHR. Estonia. Parliamentary Elections 6 March 2011 OSCE/ODIHR Election Assessment Mission Report, 2011.
20. Märt Rask, Jüri Põld, and Harri Salmann. Decision of Supreme Court 3-4-1-10-11, March 31 2011. In Estonian, `http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad`.
21. Märt Rask, Jüri Põld, and Harri Salmann. Decision of Supreme Court 3-4-1-4-11, March 21 2011. In Estonian, `http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad`.
22. Märt Rask, Jüri Põld, and Harri Salmann. Regulation of Supreme Court 3-4-1-6-11, March 23 2011. In Estonian, `http://www.vvk.ee/valimiste-korraldamine/vabariigi-valimiskomisjon-yld/kirjad`.
23. Tanel Tammet and Hannu Krosing. E-valimised Eesti Vabariigis: võimaluste analüüs (E-voting in Estonia: feasibility study), 2001. Analysis ordered by Estonian Ministry of Transport and Communications. In Estonian.