Practical Security Analysis of E-voting Systems

Ahto Buldas[1,2,3,⋆] and Triinu Mägi[4]

[1] Cybernetica AS. Akadeemia tee 21, 12618 Tallinn, Estonia.
[2] Tallinn University of Technology, Raja 15, 12618 Tallinn, Estonia.
[3] University of Tartu, Liivi 2, 50409 Tartu, Estonia. ahto.buldas@ut.ee
[4] Estonian Ministry of Justice, Center of Registers and Information Systems. Lõkke 4, 19081 Tallinn, Estonia. triinu.magi@just.ee

Abstract. We adapt game theoretic methods for studying the security of two e-voting systems: the Estonian E-Voting System (EstEVS) and Secure Electronic Registration and Voting Experiment (SERVE) performed in the United States of America. While these two systems are quite similar from technical side, security experts have made totally different decisions about their security—EstEVS was indeed used in practical elections while SERVE was decided to be insecure. The aim of this work is to clarify if the minor technical differences between these two systems were indeed a sufficient reason to distinguish between their security. Our analysis is oriented to practical security against large-scale attacks. We define a model for the real-life environment in which voting takes place and analyze the behavior of adversaries. We show that in our model EstEVS is secure and SERVE is not. The reliability of the results is still questionable because of our limited knowledge about many of the parameters. It turns out though that our main results are quite robust with respect to the choice of parameters.

1 Introduction

Many of us have dealt with electronic commerce transactions. This is already a part of everyday life. However, e-voting is not yet so widely used. A secure electronic voting system is still one of the most challenging tasks, because of the need for finding a trade-off between seemingly contradictory requirements like privacy vs. auditability. Thereby, it is difficult to adopt ordinary mechanisms of e-commerce. For example, in e-commerce there is always a possibility to dispute about the content of transactions. Buyers get receipts to prove their participation in transactions. E-voters, in turn, must not get any receipts, because this would enable voters to sell their votes.

In 2003, Estonia initiated the development of an e-voting system (further referred to as Estonian E-Voting System: EstEVS) [12]. The aim was to use e-voting in the elections of the local government councils in 2005. In January 2004, a group of American security experts revealed the security report of Secure Electronic Registration and Voting Experiment (SERVE) [1]. The SERVE system was planned for deployment in the 2004 primary and general elections and allows eligible voters to vote electronically via Internet. After examining the security of SERVE, the group of security experts recommended that SERVE should be shut down. They also declared that they do not believe

⋆ Partially supported by Estonian SF grant no. 7081, and by EU FP6-15964: “AEOLUS”.

that differently constituted projects could be more secure than SERVE. Their conclusion was that the real barriers to success in e-voting are not skills, resources, etc; it is the fact that given the current Internet and PC security technology, e-voting is an essentially impossible task. The SERVE project was indeed terminated in January 2004. At the same time, Estonia continued to develop an e-voting system and implemented it according to the plans. In their security analysis [2] estonian experts declared that EstEVS is sufficiently secure in practice.

This contradicting situation was the main initiator of this work. From closer view, both security reports are consistent and contain truthful and convincing arguments. One of the main reasons for two totally different decisions was the lack of unified rational security analysis in both reports. Some of the arguments were quite emotional, being based on experts' subjective opinions and "common wisdom". The aim of this work is to adapt rational security analysis methods for studying the two e-voting systems. It gives us the possibility to compare practical security levels of these systems.

One of the rational approaches of security is known from theoretical cryptography: security reductions, which are proofs that security conditions held under certain combinatorial assumptions, such as hardness of factoring or Diffie-Hellman problem. For estimating practical security, we also need empirical assumptions about the real world. Moreover, in theoretical cryptography the adversaries are considered to be Turing machines, which are well-defined and relatively easy to study. The real world adversaries are human beings with unpredictable behavior and different motives. Hence, for analyzing practical security, we need models for real world adversaries. In this work, we adapt *multi-parameter attack trees* [3] for analyzing the security of e-voting systems.

Real-world security is not just a technical issue. In many cases, it would be more beneficial for an adversary to bribe employees of organizations rather than to break into their computer system from outside. Hence, the model for real-life environment must consider many "social parameters" like the costs of bribing people. We create a model for real-life environment in which these parameters are accounted.

We show that EstEVS is practically secure in our model but SERVE has vulnerabilities, which make certain voting-specific attacks possible. Additionally, we show that reasonable changes in the model will not change the results of the analysis. This means that if our environment model indeed reflects the reality, then EstEVS is more secure than SERVE and the security experts' opinions were reasonable. It turns out that the main technical disadvantages of SERVE are: (1) *ballot decryption in e-voting servers*, (2) *lack of independent audit log systems*, (3) *online votes counting server that contains, besides votes, also the identities of voters*, (4) *ballots are not signed by voters*.

We tried to choose the parameters of the model so that they were as close as possible to real society. We used information from Internet, research reports, interviews with public prosecutors and well-studied attack scenarios. In spite of that, our model is obviously not perfect—the estimation of environment characteristics is quite subjective. Still, this work emphasizes the need for better measurements of these environment characteristics, in case we have to analyze the practical security of e-voting systems. Better measurements definitely would improve this security analysis. Unfortunately, it was not possible to include all details of the analysis into this paper. A somewhat more complete representation can be found in the master thesis of Triinu Mgi [13].

2 Security Properties of e-voting

High security is essential to elections. Democracy relies on broad confidence in the integrity of elections. There has been a lot of attention to electronic voting by cryptographers because of the challenging need to simultaneously achieve many seemingly contradictory properties, like privacy, auditability, and correctness. The most important requirements of e-voting are the following:

i. Eligible voters are able to cast ballots that are counted in the final tally.
ii. Non-eligible voters are disfranchised.
iii. Eligible voters are unable to cast two ballots that are both counted in the final tally.
iv. Voting is private and incoercible. This apparently contradicts correctness, because eligible voters must be identified to distinguish them from non-eligible ones.
v. It is possible for auditors to check whether the final tally is correctly computed. This requirement says that a group of dedicated auditors or Electoral Committee can check the correctness of voting.
vi. The results of voting must be secret until the official end of voting. No one, including votes' counting officers, must be able to reveal the final tally before the official date. Otherwise, the result of voting could affect voters' decisions.

Some researches suggest stronger security properties of e-voting but we concentrate only to the most important properties that directly correspond to the requirements of traditional voting. One of the main starting points of this work is that the security of e-voting should be comparable to that of traditional voting, though we might achieve more by using contemporary cryptographic techniques. The properties listed above are relevant for almost all voting systems and they are the the basis of our security analysis.

For securely implementing e-voting systems in real-life elections cryptographic schemes are clearly not the main problem. A far deeper concern is whether the workstations of "average citizens" (in which computer viruses are everyday visitors) can be used for such a security-critical task.

3 State of the Art

Internet voting systems have been implemented in Europe in couple of places, for example in the Netherlands in 2004 in the European Parliamentary elections. The target group consisted of the Dutch electors' resident abroad and electors resident in the Netherlands who are temporarily abroad on the Election Day. In Great Britain, remote electronic voting systems were used in the local elections of 30 municipalities in 2003.

In the United States of America, many attempts have been made to use e-voting systems. The Voting over the Internet (VOI) project was used in the general elections of 2000 in four states. The Internet votes were legally accepted, but their amount was small (84 votes) [11]. VOI's experiment was too small for being a likely target of attacks.

Another e-voting project named Secure Electronic Registration and Voting Experiment (SERVE) was developed for primary and general elections in 2004 in the United States of America. The eligible voters of SERVE were mainly overseas voters and military personnel. The US Department of Defense terminated SERVE in 2004 because a group of security experts had found that SERVE was not sufficiently secure.

The Estonian e-voting system was applied first time in the municipal elections in 2005. The second implementation was in 2007 in Parliamentary elections. There were 5.4 per cent of e-votes among all votes.

4 Description of e-voting Systems

In the following, we describe EstEVS and SERVE and emphasize their main differences. The Estonian e-voting system is implemented from the sixth day up to the fourth day before the Election Day. There are two main principles in EstEVS.

(1) Each eligible voter is able to re-vote, so that the older votes are deleted.
(2) Traditional voting cancels electronic votes.

In EstEVS the national Public Key Infrastructure is applied and voters use their authentication and digital signature certificates for casting votes. In SERVE, it is possible to vote any time within 30 days before the Election Day until the closing time of polls on the Election Day. Every voter can vote only once. There are no Public Key Infrastructure and ID-cards used in SERVE. In both e-voting systems if considerable attacks against e-voting have been detected, Electoral Committee might stop e-voting and cancel the result of voting. In general terms, e-voting systems consist into four main components:

– *Voter Applications* - a web application for casting votes.
– *Network Sever* - an server that provides voters an interface for casting their votes.
– *Votes Storing Server* - an server for storing, managing, and maintaining votes.
– *Votes Counting Server* - a server for counting the final tally.

In SERVE, Votes Counting Server is online while in EstEVS it is off-line. Additionally, EstEVS has an independent audit log system which consists of traces of all voting procedures. All log records are cryptographically linked. Log files enable to audit the e-voting system. SERVE has a similar architecture to that of EstEVS, except the log files system and the off-line Votes Counting Server.

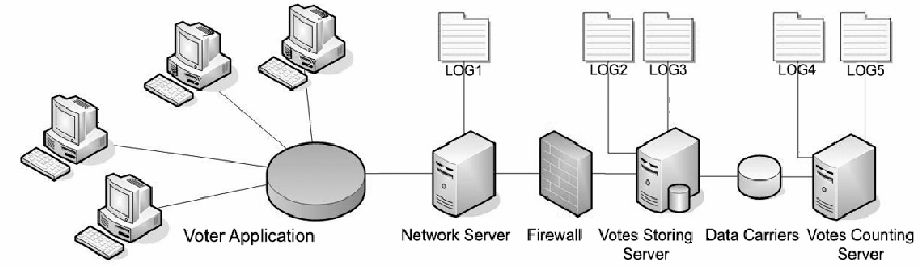

Fig. 1. Components of the Estonian e-voting system.

We now briefly describe the processes of e-voting in EstEVS and SERVE. Fig. 1 depicts the components of the EstEVS. Voting procedure is started with a voter connecting to Network Server via the SSL protocol. Voters enter their personal data for authentication. In EstEVS, national Public Key Infrastructure is applied and voters use their authentication certificates. In SERVE, there is a voters' registration process before the e-voting and voters authenticate themselves with passwords. When the connection is established, then a signed ActiveX control is downloaded to voter's computer in both e-voting systems. An authenticated voter makes his/her choice from a list of candidates transferred from Network Server. In EstEVS the application encrypts the vote by using the public key of Votes Counting Server, however in SERVE the application use the public key of Votes' Storing Server for encrypting the votes.

In SERVE, Voter Application sends an encrypted ballot and voter's personal data to Network Server, which forwards the encrypted ballot and voter's personal data to Votes Storing Server. In EstEVS, voters sign the encrypted ballots with their digital signature certificates. Network Server checks whether the session owner is the same person who signed the encrypted ballot (via ID-card authentication) and in case of positive acknowledgment, transfers the signed and encrypted ballot to Votes Storing Server.

Votes Storing Servers verify voter's franchise and if the voter had already voted. The systems reply to each correctly cast vote with a textual receipt. In EstEVS, after the end of the e-voting period Votes Storing Server cancels multiple ballots and saves the trace of canceled ballots into the log file system. Next, the server separates digital signatures and encrypted ballots. In SERVE, Votes Storing Server *decrypts* the ballots, and separates ballots from personal data. After that, Votes Storing Server encrypts the ballots again without voters' personal data with the public key of Votes Counting Server.

In SERVE, Votes Counting Server downloads the list of voters and the encrypted ballots from Votes Storing Server when Votes Counting Server updates its database. In EstEVS, encrypted ballots are transferred to the off-line Votes Counting Server by using data carriers. For counting votes, Votes Counting Server decrypts the encrypted ballots by using the private key of Votes Counting Server. Only accepted format of votes are counted to the final tally. In EstEVS, Votes Counting Server outputs the final tally and in the SERVE system it outputs the final tally and the list of voters. Table 1 depicts the main differences between the two systems.

Table 1. Differences between the two e-voting systems.

Characteristic	**EstEVS**	**SERVE**
e-voting used on the Election Day	No	Yes
Possibility to re-vote at the polling station	Yes	No
National Public Key Infrastructure	Yes	No
Voters sign the ballots	Yes	No
State of votes in Votes Storing Server	Encrypted	Not encrypted
State of Votes Counting Server	Off-line	On-line
Audit log system	Yes	No

5 Analysis Method

To measure the real security of e-voting, we should analyze the security in an objective way. It would be insufficient (at least for the purposes of this work) to rely on subjective opinions of security experts—we try to put their opinions to solid ground by providing them with a method to determine whether the system is secure.

In order to declare that e-voting system is secure it must be as secure as traditional voting, which is considered to be practically secure and resistant to *large-scale threats*. This means that the e-voting systems must also be secure against the large-scale voting-specific attacks. A large-scale attack may cause considerable changes in the final tally or reveal large number of votes. Therefore, for estimating practical security of the systems, we try to create an environment model as close as possible to the real-life environment in which e-voting systems are used. In addition to technological parameters, we have to make assumptions about society, people, and motives of attackers. We assume that adversaries are gain-oriented and attack on purpose—to affect the result of elections. We analyze adversarial behavior by using the game-theoretical setting suggested in [3]. According to this setting, attacks are viewed as games the profitability of which (for attackers) depends on the following parameters of the environment model:

- Gains - the gains of the attacker, in case the attack succeeds.
- Costs - the cost of the attack.
- p - the success probability of the attack.
- q - the probability of getting caught (if the attack was successful).
- $\mathsf{Penalties}$ - the penalties in case the attacker is caught (if the attack was successful).
- $q_{\_}$ - the probability of getting caught (in case the attack was not successful).
- $\mathsf{Penalties}_{\_}$ - penalties if the attacker was caught and the attack was unsuccessful.
- $\mathsf{Outcome}$ - average outcome of an attacker.

Considering all these parameters, rational attackers calculate the expected outcome of the game, which determines their decision about whether to attack or not:

$$\mathsf{Outcome} = -\mathsf{Costs} + p \cdot (\mathsf{Gains} - q \cdot \mathsf{Penalties}) - (1-p) \cdot q_{\_} \cdot \mathsf{Penalties}_{\_} \ .$$

Attackers do not attack, if the outcome of the attack-game is negative and they always choose the most profitable ways for attacking. For the sake of simplicity we denote:

- by π the average penalty if the attack was successful, i.e. $\pi = q \cdot \mathsf{Penalties}$;
- by $\pi_{\_}$ the average penalty if the attack was unsuccessful, i.e. the outcome is equal to $-\mathsf{Costs} + p \cdot (\mathsf{Gains} - \pi) - (1-p) \cdot \pi_{\_}$.

For better estimation of the parameters, attacks are split into simpler ones by two rules. *AND-rule* states that the component-attacks are all necessary for the original attack, whereas *OR-rule* states that at least one of the components is needed for the original attack. Such a decomposition procedure is iterated until we can estimate the parameters of all components, i.e. they can be deduced from our model of environment. The composition tree that corresponds to this process is called an *attack tree* [5]. Each node of an attack tree represents an attack. Leaf nodes represent atomic attacks for which all parameters are known. For simplicity, it is assumed [3] that Gains is the same for all nodes. To compute the parameters of the root node, we need the following rules [3]:

– For an OR-node with children $(\mathsf{Costs}_1, p_1, \pi_1, \pi_{1-})$ and $(\mathsf{Costs}_2, p_2, \pi_2, \pi_{2-})$ the parameters $(\mathsf{Costs}, p, \pi, \pi_-)$ are computed as follows:

$$(\mathsf{Costs}, p, \pi, \pi_-) = \begin{cases} (\mathsf{Costs}_1, p_1, \pi_1, \pi_{1-}), \text{ if } \mathsf{Outcome}_1 > \mathsf{Outcome}_2 \\ (\mathsf{Costs}_2, p_2, \pi_2, \pi_{2-}), \text{ if } \mathsf{Outcome}_1 \leq \mathsf{Outcome}_2 \end{cases},$$

where $\mathsf{Outcome}_i = -\mathsf{Costs}_i + p_i \cdot \mathsf{Gains} - p_i \cdot \pi_i - (1 - p_i) \cdot \pi_{i-}$ for $i = 1, 2$.

– For a AND-node with children $(\mathsf{Costs}_1, p_1, \pi_1, \pi_{1-})$ and $(\mathsf{Costs}_2, p_2, \pi_2, \pi_{2-})$ the parameters $(\mathsf{Costs}, p, \pi, \pi_-)$ are defined as follows (where $\tilde{p_i}$ will denote $1 - p_i$):

$$\mathsf{Costs} = \mathsf{Costs}_1 + \mathsf{Costs}_2; \qquad p = p_1 \cdot p_2; \qquad \pi = \pi_1 + \pi_2; \quad \text{and}$$
$$\pi_- = \frac{p_1\tilde{p_2}(\pi_1 + \pi_{2-}) + \tilde{p_1}p_2(\pi_{1-} + \pi_2) + \tilde{p_1}\tilde{p_2}(\pi_{1-} + \pi_{2-})}{1 - p_1p_2}.$$

6 Adversarial Model and Threats

In our analysis, we consider adversaries as a rationally thinking persons who always choose the most profitable attacks and who will not attack if all possible attacks are unprofitable. We do not model adversaries as inside attackers. We assume that the development team of e-voting has been created carefully and the team members are benevolent by themselves. However, we assume that the team members can be influenced from outside (for example, bribing) in order to affect an e-voting system maliciously. In this work, we do not analyze crimes against person because they pose an equal threat to traditional voting. Hence, we do not consider that anybody is involved in an attack by coercion or violence. We analyze the behavior of attackers through the components of e-voting systems by using multi-parameter attack trees [3]. For example an adversary has the following activities:

– to attack Voter Application in order to affect the votes' casting process;
– to attack the connection between Voter Application and Network Server in order to affect the votes before they are received by Network Server;
– to attack Network Server in order to affect the votes' reception.

By using the activities that are allowed in the adversarial model, the aim of attacker is to perform a large-scale attack. Small-scale attacks, which affect a small number of votes, do not affect the overall result of voting and hence do not pose a threat to democratic society as a whole. So, we will study large-scale attacks that cause considerable changes in the final tally or a large scale of votes to become revealed.

How big is *large number*? How many votes should be changed or revealed in e-voting systems so that we may talk about a large-scale attack? For estimating this parameter we analyzed elections in Estonia and in the United States of America. We saw that the minimum average per cent of votes to affect the result of voting could be 4 per cent [4, 9]. There has been an exception in the presidential elections of the USA in 2004. The difference between the rates of parties was only 0.0246. The number of target voters of EstEVS and of SERVE was 1 million and 6 millions, respectively. Obviously, one hundred infected computers do not affect the overall result of elections. If 1,000

computers are infected, it would be possible to affect 0.1 per cent of the Estonian votes and 0.016 per cent of the United States votes. To summarize, we consider that infecting 1,000 computers is sufficient for having a large-scale attack in e-voting systems.

We say that e-voting is *practically secure* if it resists the following large-scale attacks:

1) *Large-scale votes' theft.* The aim of the attack is to change votes or to give more (forged) votes for favorite candidates. Such an attack is possible only if the adversary is able to cast ballots in the name of many users or the system enables voters to cast multiple ballots that are all counted in the final tally.
2) *Large-scale disfranchisement of votes.* It means that a large number of correctly encrypted ballots from eligible voters never reach Votes Storing Server. Attack could also selectively disfranchise eligible votes in order to eliminate undesirable ones. Note that the aim of a rational (and well-prepared) attacker is not to cause the overall failure of e-voting and hence such an attack should stay unnoticed.
3) *Large-scale votes' buying and selling.* It means that a large number of votes are sold. The aim of the attack is to increase the amount of votes for certain candidates.
4) *Large-scale privacy violation.* The aim of the attack is to reveal how voters have voted. This my cause violence and persecution in the society.

7 Security Assumptions

We make several simplifying assumptions which will eliminate many irrelevant details of our empirical analysis and thereby keeps the "big picture" of the analysis observable to the reader. For example, we assume that the cryptographic schemes used in e-voting systems are secure. Our analysis uses the following assumptions:

- *Assumption I*: It is impossible to forge signatures without private keys.
- *Assumption II*: It is impossible to deduce votes from encrypted ballots.
- *Assumption III*: Adversaries do not have access to the private keys of voting servers. Key management at the server side is sufficient to prevent key compromise.
- *Assumption IV*: Voters' registration is secure. EstEVS uses national PKI and does not need voters' registration—ID-cards with authentication and digital signature certificates are issued to all citizens. In this work, we assume that sharing of authentication data and digital signature certification is secure in EstEVS. For fair comparison of the two systems, we also assume that the phase of voter's registration in SERVE is secure.
- *Assumption V*: The phase of votes' counting behaves as specified. All correctly cast votes that are received by Votes Counting Server are counted correctly. This assumption might be unjustified in voting systems, because the insider threats are even more common than the outsider threats. However, in this analysis the insider threats of votes' counting phase are not taken into account.
- *Assumption VI*: The log file system of EstEVS is secure. All records in audit logs are cryptographically linked and it is impossible to modify them without detection.
- *Assumption VII*: If considerable attacks are detected that cause misbehavior of e-voting then e-voting is immediately stopped and the results of e-voting canceled. Both EstEVS and the SERVE project have justified this property in the requirements of the systems. This is decided by court or Electoral Committee.

8 Model of Environment

A meaningful comparison of two systems must be based on equal or comparable benchmarks. Hence, we create the same environment for the both e-voting systems. It is clear, that the environments of EstEVS and SERVE are different in real life. Moreover, it is even hard to describe these environments adequately and give real characteristics of environment. For example, it is hard to estimate what is the probability of catching and convicting attackers, if voters deliberately create connections to an actively compromised voting server. For adequately specifying the characteristics of an environment for e-voting systems, it is necessary to study the motives and purposes of attacks, success probabilities of attacks, detection probabilities of attacks, awareness of computers' users, punishments for cyber-crimes, etc. In order to make rational decisions about practical security of e-voting systems, we have to know these parameters with sufficient accuracy. Note that if we are unable to do it, then this would also mean that *we do not know whether these systems are secure*. Hence, the way to go here is to obtain better estimates for these parameters.

We create a hypothetical environment for analyzing security of the two e-voting systems. We try to estimate the parameters of the environment as close as possible to the real society. For estimating these parameters we have used information from Internet, from research papers, interviews with specialists and typical attacking scenarios. We assume that typical attackers do not make extensive social research for getting information on whether it is profitable to attack. Quite probably, gain-oriented attackers would analyze the same information from Internet and make decisions intuitively. Definitely, this hypothetical environment is not perfect, but it is the best we can do for comparing the security of the two e-voting systems. Our model contains assumptions about: (1) society, (2) people, (3) technical vulnerabilities, and (4) detection. These assumptions are commented in the following subsections.

8.1 Assumptions about Society

Voting is a fundamental tool of democracy and one of the main rights in democratic society. We assume that the environment we model is a well-developed democratic society in which *the aim of crime determines the seriousness of crime*. If the aim of the crime is to affect the result of voting then it is viewed as a serious crime against society, no matter how it was performed and whether the crime was "technically successful". Hence, we assume that $\mathsf{Penalties} \approx \mathsf{Penalties}_{\_}$. Moreover, the punishment for crime is at least dispossession of the gains obtained from the crime. Thus, we assume that $\mathsf{Gains} \leq \mathsf{Penalties}$. For simplicity, we study the limit case $\mathsf{Penalties} \approx \mathsf{Gains} \approx \mathsf{Penalties}_{\_}$, which implies

$$\mathsf{Outcome} \approx -\mathsf{Costs} + \mathsf{Gains} \cdot [p \cdot (1-q) - (1-p) \cdot q_{\_}] \ .$$

Parties spend lots of money for campaigns of election. Probably, the gain is even bigger. In Estonia, parties spend about $2 million [10] for a campaign of election. We assume that Gains of affecting the result of election is at least 5 times bigger, so $10 million.

By the data available in Internet the price of obtaining malicious code is about $50. A person can be bribed for about $50,000 [7]. We assume that attackers are rationally

and economically thinking. Hence, to calculate the cost of attack, we focus on self-cost. Even, if the price of developing a forged Network Server is $2 million, the expenses of attacks are small compared to the gains.

Considering the specificity of elections, Costs are always much smaller than Gains. Hence, the value of Costs does not affect attacker's final decision to attack an e-voting system or not. Therefore, we may even assume that $\mathsf{Costs} \approx 0$. If the e-voting system is secure when $\mathsf{Costs} = 0$ then the system is also secure when $\mathsf{Costs} > 0$. Therefore, under these simplification we conclude that

$$\mathsf{Outcome} \approx -\mathsf{Costs} + \mathsf{Gains} \cdot [p \cdot (1-q) - (1-p) \cdot q_{-}] < 0 \ ,$$

whenever $p \cdot (1-q) - (1-p) \cdot q_{-} < 0$. To summarize, considering the particularity of e-voting we may estimate only three parameters p, q and q_{-} of the attack game for estimating the profitability of attacks. In the following we list the characteristic probabilities (Char. 1-15) of the environment that we use in our analysis:

8.2 Characteristic Probabilities

In the following, we list 15 characteristic probabilities of the environment that we use in our security analysis. These probabilities are divided into assumptions about: (1) people (Char. 1-7), (2) technical vulnerabilities (Char. 8-11), and (3) detection (Char. 12-15).

Char. 1. About 1 per cent of voters will notice that their computers are infected and will inform the authorities about it. Thereby, the success probability of attacking large number (1000) of voters' workstations (without this being noticed) is $p \leq 0.99^{1000}$.

Char. 2. At least 1 per cent of electronic voters verify the authenticity of the Network Server certificate, the signature of ActiveX component and wait for the confirmation of e-voting. We assume that if a voter is aware of the need to verify the certificate of Network Server, then he is also aware of the need to verify the signature of ActiveX component and to wait for the confirmation about accepted vote. The probability that 1,000 voters do not verify the certificate of Network Server or the signature of ActiveX component or do not wait for the signed confirmation from the e-voting server is $p \leq 0.99^{1000}$. Such a modeling of voters is somewhat idealistic, because all voters are assumed to have the same values of probability. In practice, the attacker may estimate these values by guessing the technical skills and carefulness of the voters and then to attack those with lower skill and careless.

Char. 3. About 33 per cent of people can be bribed for $50,000 [7].

Char. 4. The probability that voters click on a (well-created) malicious link is ≈ 0.6. Hence, the probability that a fixed set of 1000 people will use the link is $p \leq 0.6^{1000}$.

Char. 5. About 1 per cent of people involved in attacks will reveal information that causes the attackers to be caught. Hence, the probability that a group of 10 people will get caught is $q \geq 1 - 0.99^{10} \approx 0.096$.

Char. 6. We assume intuitively that voter would sell his vote with probability 0.5 by using active votes' selling environment. The probability that voter would sell the vote

by using more anonymous ways is 0.7. It means that a voter would feel more secure to participate in a scheme of votes' selling and buying by using computer based voting data saving and proving software.

Char. 7. The probability that voters agree to vote many times (for an attacker) is 0.9.

Char. 8. The probability of exploiting a bug in an operating system or hardware and getting access to a system is ≈ 0.002. We assume that bugs in operating systems or in hardware are discovered once in 3 years on average. Within 2 days, viruses can exploit the bug. Within 7 days, there will be countermeasures available. Hence, attackers have one week per three years to exploit the bug. Thereby, at every moment, there is bug to exploit with probability 0.0064. The probability of getting unauthorized access to administrative areas of a system or to other internal modules is 0.21 [8]. Hence, the probability of exploiting a bug and getting access to the system is $0.0064 \cdot 0.21 \leq 0.002$.

Char. 9. The probability that a forged Network Server or malicious code succeeds in attack is $p \approx 0.95$. Usually, the accordance between functions of developed information system and claimed system requirements is not 95 per cent. However, for estimating the security of system we promote attackers. If the system is secure against powerful and penetrating attacks, then it is secure against weaker attacks.

Char. 10. The probability that voters' computers are vulnerable is about 0.31 [6].

Char. 11. The probability that adversaries have succeeded to gain control over the connection between the e-voting servers is 0.15. We assume intuitively that if the probability that voters' computers are vulnerable for session controlling is 0.31 [6], then the control over the session between servers is harder at least twice as hard, i.e. $p \leq 0.15$.

Char. 12. Code review and auditing can detect about 30% of software errors.

Char. 13. Bribing that causes damage is detected with probability $q \leq 0.3$ [7].

Char. 14. Attacks against insecure server conf. are detected with probability 0.05.

Char. 15. The probability that a successful crime against the e-voting system will be convicted is 0.8. Unsuccessful crimes will be convicted with probability 0.2 [7].

9 Attack Game Analysis

First, we decompose the four large-scale attacks (listed in Sec. 6) by using the OR-rule, i.e. we created a list of alternative ways of attacking the two e-voting systems. After that, we analyzed all alternatives separately by using the security assumptions (Sec. 7), the environment characteristics (Sec. 8), and the attack tree method (Sec. 5). Table 2 depicts the decomposition of the large-scale attacks.[5] For example, we studied seven alternatives for large-scale votes' theft and four alternatives (so called *sub-attacks*) for large-scale disfranchisement attack. For the lack of space, we will not present detailed analysis of all possible attacks and alternatives. First, we focus on a few sub-attacks and then present a more complete analysis for the *large-scale votes' buying attack*.

[5] The table is not complete and does not contain decomposition of all four large scale attacks.

Table 2. Sub attacks for large-scale voting specific attacks. By '-' we mean that the sub-attack is impossible or insufficient.

Attack	Sub-attacks	EstEVS	SERVE
Large-scale votes' theft	Large-scale control over voters' processes	unprofitable	unprofitable
	Large-scale access to voters' private keys	unprofitable	unprofitable
	Eligible voters cast votes more than once	unprofitable	unprofitable
	Large-scale disfranchisement in two servers	unprofitable	-
	Large-scale modification of ballots in the connection between Voter Application and Network Server	-	unprofitable
	Control over processes of Votes Storing Server	-	profitable
	Large-scale votes' adding in Votes Counting Server	-	unprofitable
Large-scale disfranchisement of votes	Large-scale control over voter processes	unprofitable	unprofitable
	Large-scale disfr. before receiving votes	unprofitable	unprofitable
	Large-scale disfr. in two servers	unprofitable	unprofitable
	Control over processes of Votes Storing Server	-	profitable
Large-scale votes' buying/selling	(decomposition omitted)	unprofitable	profitable
Large-scale privacy violation	(decomposition omitted)	unprofitable	profitable

Large-scale control over voters' processes. In EstEVS and in SERVE, large-scale control over voters' processes is possible either by infecting computers one-by-one or by using automatically propagating attacking software (viruses etc.). We assume that both methods have the same expenses. By assumptions, with probability $p \leq 0.99^{1000}$ attackers are able to smuggle malicious code into voters' computers and get the desired data by Char. 1. A large-scale access to voters' private keys is a serious attack and the estimation of detecting the attack is 0.8 by Char. 15. If we assume that the attack was not successful, then the probability of getting caught is $q \geq 0.096$ by Char. 5. For estimating the profitability, we compute $\mathsf{Outcome}$ as follows:

$$\begin{aligned}\mathsf{Outcome} &\leq -\mathsf{Costs}+\mathsf{Gains}\cdot[p\cdot(1-q)-(1-p)\cdot q_{-}]\\ &= -\mathsf{Costs}+\mathsf{Gains}\cdot[0.99^{1000}\cdot(1-0.8)-(1-0.99^{1000})\cdot 0.096]\\ &< -\mathsf{Costs}-\mathsf{Gains}\cdot 0.096 < 0\ .\end{aligned}$$

As $\mathsf{Gains} \gg \mathsf{Costs}$, the value of Costs does not affect the attacker's final decision. The attack is unprofitable, if $p(1-q)-(1-p)q_{-} < 0$. Additionally, even if the probabilities q and q_{-} of getting caught are 0.096, the attack is not profitable. Therefore, an attack via large-scale control over the voters' processes is unprofitable in both systems.

Large-scale access to voters' private keys. An average voter is unable to keep its own workstation secure enough to exclude all possible abuses of the private key. For example, adversaries can steal voter's password for activating the ID-card. Still, it is not possible to arrange a large-scale theft of cards, because voters would notice it immediately and elections will be canceled by Assumption VII. The success probability of large-scale access to voters' private keys is $p \leq 0.99^{1000}$ by Char. 1. The argumentations used here are similar to the previously analyzed large-scale attack. Therefore, large-scale access to the voters' private keys is unprofitable for rational attackers in both e-voting systems.

Large scale votes' buying. Large-scale buying and selling of votes is possible only if there is a possibility to prove a vote. In case the voter could not prove how he/she

had voted, the votes' buying and selling is not a trustful deal. There is a theoretical advantage for adversaries in the e-voting systems compared to adversaries in traditional voting. The adversaries do not have to physically contact with every voter for affecting his choice. The adversaries should affect at least 1,000 voters for affecting the result of e-voting. Obviously, the easiest way to affect many voters is to offer votes' buying and selling services. In Section 8, we assumed that Gains of an attack could be $10 million. Let us analyze, whether it is possible to eliminate the parameter Costs like we did previously. Obviously, the price of organizing and preparing the attack is much smaller than Gains. The biggest expense is the price of votes. In the case when adversaries spend 20 per cent of the profit for buying 1,000 votes, the price of vote is $2,000. We assume that such price is attractive for vote sellers. Therefore, Costs for buying at least 1,000 votes is smaller than Gains. In the following, we create attack trees for large scale votes' buying in SERVE and in EstEVS.

9.1 Analysis of SERVE

An attack tree for large-scale votes' buying in SERVE is depicted in Fig. 2 (left) and the computations in Table 3. There are three possibilities to arrange votes' buying and selling in SERVE. First, by using votes selling and buying web server (Sub tree A). Voters connect to votes buying server for casting their votes. The server saves voters' choices and sends ballots to Network Server. Second, voters use votes saving software for getting the receipt of voting and cast a vote directly to Network Server (Sub tree B). A receipt consists of voter's data, a vote, a random number and an encrypted ballot. The voters send the receipts to the adversary for proving how they voted. The adversary attacks the e-voting server for getting a proof that a ballot is received. For inserting malicious code into servers there are four possibilities: software developer of a server is bribed (B.3.2.1.), server administrator is bribed (B.3.2.2.), insecure configuration management is exploited (B.3.2.3.). Third, the adversary attacks the servers of e-voting for checking how voters voted (Sub tree C). Votes Storing Server of SERVE decrypts the ballots. Adversary attacks against Votes Storing Server for the purpose of stealing pairs of voters' data and ballots. These pairs give a proof how voters voted. In the following, we analyze these sub-trees.

Sub-tree A: With probability 0.95 votes' buying and selling information system is developed successfully by Char. 9. To consider the active and public attack, the probability of detecting the attacking group is 0.8 by Char. 15. For analyzing voters connection to votes' buying server, we assume that that 50 per cent of voters would sell their vote by Char. 6. The probability of detecting voters who have voted by using votes buying server is 0.8 because this is the probability of detecting the votes buying server. To summarize, it is not profitable for attacker to attack through votes' buying and selling server.

Sub-tree B: The probability of the votes' saving software functioning correctly is 0.95 by Char. 9. The probability of detecting the votes' saving software is 0.096 by Char. 5. The success probability of voters using the software is $p = 0.7$ by Char. 6. If there are at least 1,000 people involved and Char. 5 is justified then the probability of detection and punishment of saving the receipt is $q = q_{-} = 1 - 0.99^{1000}$. The probability of

the malicious code successfully getting voters' data and encrypted ballots from a voting server is 0.95 by Char. 9. The detection probability is 0.096 by Char. 5. According to Char. 3, a software developer and a server administrator are bribed with probability 0.33. Based on the assumption that development teams use code reviews, misbehavior in software is detected with probability 0.3 by Char. 12. Therefore, for estimating the probability of a software developer getting caught, we consider information leaking and the detection rate of misbehavior in server. Hence, the probability of getting caught without succeeding is $q_{-} = 0.096 + 0.3 = 0.396$ by Char. 5 and Char. 12. Bribery is detected with probability 0.3 by Char. 13. The success probability of detecting a software developer is $q \approx 0.096 + 0.3 + 0.3 \approx 0.7$ by Char. 5, Char. 12 and Char. 13. In the event that the attack was not successful, the probability of detecting that the server administrator was bribed is at least $q_{-} \geq 0.096$ by Char. 5. Considering the value of q_{-} and Char. 13, the probability of a server administrator being caught is $q \approx 0.096 + 0.3 \approx 0.4$. Insecure configuration management is successfully exploited with probability $p \leq 0.002$ by Char. 8. We assume intuitively that the probability of detection of the exploiting configuration management is 0.05 by Char. 14. Control over the connection between servers is successful with probability 0.15 and the probability of the detection of the attack is 0.096 by Char. 11 and Char. 5. To summarize, spreading votes' receipt software does not give a profitable attack in our model.

Sub tree C: The analysis of sub-tree is analogous to the analysis of sub-tree B.3. Attacking Votes Storing Server for getting voters' ballots is successful with probability $p \approx 0.32$ and it has positive Outcome of the attack game. Therefore, Large-scale votes' buying in SERVE is profitable, considering our model.

Table 3. Large-scale votes' buying in SERVE.

Node	**Description of attack**	**Type**	p	q	q_{-}	π	π_{-}	**Outcome**
A	Votes buying server.	AND	0.475	0.64	0.64	$1.6 \cdot 10^7$	$8.7 \cdot 10^6$	$-7.45 \cdot 10^6$
A.1	Attacking software is developed.		0.95	0.8	0.8	$8.0 \cdot 10^6$	$8.0 \cdot 10^6$	$1.5 \cdot 10^6$
A.2	Voters connect to the server.		0.5	0.8	0.8	$8.0 \cdot 10^6$	$8.0 \cdot 10^6$	$-3.0 \cdot 10^6$
B	Spreading votes' receipt software.	AND	0.208	0.0037	0.00089	$1.59 \cdot 10^7$	$1.42 \cdot 10^7$	$-1.23 \cdot 10^7$
B.1	Developing data-saving software.		0.95	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$8.54 \cdot 10^6$
B.2	Voters use software to save receipts.		0.7	0.999957	0.999957	$1.0 \cdot 10^7$	$1.0 \cdot 10^7$	$-3.0 \cdot 10^6$
B.3	Obtain ballots from server.	AND	0.3135	0.0384	0.0092	$4.96 \cdot 10^6$	$1.40 \cdot 10^6$	$6.21 \cdot 10^5$
B.3.1	Developing malicious code		0.95	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$8.54 \cdot 10^6$
B.3.2	Inserting code into server.	OR	0.33	0.4	0.096	$4.0 \cdot 10^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
B.3.2.1	Software developer is bribed.		0.33	0.7	0.396	$7.0 \cdot 10^6$	$3.96 \cdot 10^6$	$-1.6 \cdot 10^6$
B.3.2.2	Server administrator is bribed.		0.33	0.4	0.096	$4 \cdot 10^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
B.3.2.3	Insecure configuration is exploited.		0.002	0.05	0.05	$5.0 \cdot 10^5$	$5.0 \cdot 10^5$	$-4.8 \cdot 10^5$
B.3.2.4	Control connections between servers.		0.15	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$5.4 \cdot 10^5$
C	Get ballots from Votes Storing Server.	AND	0.3135	0.0384	0.0092	$4.96 \cdot 10^6$	$1.40 \cdot 10^6$	$6.21 \cdot 10^5$
C.1	Develop malicious vote-saving code.		0.95	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$8.54 \cdot 10^6$
C.2	Inserting code into server.	OR	0.33	0.4	0.096	$4.0 \cdot 10^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
C.2.1	Software developer is bribed.		0.33	0.7	0.396	$7.0 \cdot 10^6$	$3.96 \cdot 10^6$	$-1.6 \cdot 10^6$
C.2.2	Server administrator is bribed.		0.33	0.4	0.096	$4 \cdot 10^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
C.2.3	Insecure configuration is exploited.		0.002	0.05	0.1	$5.0 \cdot 10^5$	$1.0 \cdot 10^6$	$-9.79 \cdot 10^5$

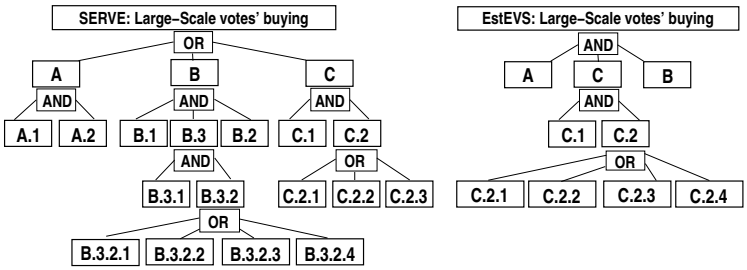

Fig. 2. Attack trees for large-scale votes' buying attack in SERVE (left) and in EstEVS (right).

9.2 Analysis of EstEVS

The attack tree Large-scale votes' buying for EstEVS is depicted in Fig. 2 (right) and the computations in Table 4. Votes buying attack against EstEVS has just one option. Adversaries develop software for saving voting data as a receipt. Voters who wish to sell their votes use the software in their computers for delivering the voting receipt. Adversaries attack an e-voting server for getting pairs of voters' data and encrypted ballots. The comparison of receipts and encrypted ballots gives the proof how voters had voted. Without getting control over one of the voting servers it would not be sure whether the ballots were really sent to the voting server.

In EstEVS, it would be impossible to sell votes via votes' buying server because Network Server verifies if the session owner is the same person who signed the ballot. Votes' buying server cannot impersonate voters without having access to their ID-cards.

In EstEVS, the ballots stay encrypted until the votes' counting phase. Hence, without the private key of Votes Counting Server it is insufficient to attack the voting server for checking how voters voted. By Assumption III, adversaries do not have the key.

The tree for votes' buying in EstEVS is similar to the sub-tree B of the corresponding tree of SERVE. Outcome of this tree is negative and hence EstEVS is secure against large-scale votes' buying in our model.

Modification of environment characteristics. For getting some more justification to our choices of parameters, we made some non-systematic robustness tests. We tried to change the environment characteristics so that the value of attack game would change its sign (from positive to negative, or vice versa). For example, we had to decrease Char 2. about 10 times for inverting a value of an attack game. It turned out that several parameters had to be changed simultaneously for inverting game values and keeping changes reasonable (10 times is clearly too much!). Also the punishment and detection characteristics had to be changed approximately 10 times in order to invert game values. So, it turned out that reasonable changes do not have much influence on the final results of our analysis, which to some extent increases our belief about the truthfulness of the results. However, limited knowledge about the real values and the embryonic state of the robustness analysis do not enable to make any conclusions about the real security of these two systems.

Table 4. Large-scale votes' buying in EstEVS.

Node	Description of attack	Type	p	q	q_-	π	π_-	Outcome
	Large-scale votes' buying.	AND	0.2085			$1.59 \cdot 10^7$	$1.42 \cdot 10^7$	-1.2510^7
A	Develop data-saving software.		0.95	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$8.54 \cdot 10^6$
B	Voters use the software.		0.7	0.999957	0.999957	$1.0 \cdot 10^7$	$1.0 \cdot 10^7$	$-3.0 \cdot 10^6$
C	Obtain ballots from voting server.	AND	0.3135	0.0384	0.0092	$4.96 \cdot 10^6$	$1.4 \cdot 10^6$	$6.21 \cdot 10^5$
C.1	Develop malicious code.		0.95	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$8.54 \cdot 10^6$
C.2	Insert the code into server.	OR	0.33	0.4	0.096	$4.0 \cdot 10^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
C.2.1	Software developer is bribed.		0.33	0.7	0.396	$7.0 \cdot 10^6$	$3.96 \cdot 10^6$	$-1.66 \cdot 10^6$
C.2.2	Server administrator is bribed.		0.33	0.4	0.096	$4.0\dot{1}0^6$	$9.6 \cdot 10^5$	$1.34 \cdot 10^6$
C.2.3	Insecure configuration is exploited.		0.002	0.05	0.05	$5.0 \cdot 10^5$	$5.0 \cdot 10^5$	$-4.8 \cdot 10^5$
C.2.4	Control the connection between servers.		0.15	0.096	0.096	$9.6 \cdot 10^5$	$9.6 \cdot 10^5$	$5.4 \cdot 10^5$

10 Further Work

The results of the work are still disputable and need further improvement and justifications, because the characteristics of the defined environment model are arguable. However, this work is one of the first attempts to rationally analyze the security of e-voting by combining both the technical and the social aspects, which are all necessary for making any reliable decisions about the real security of e-voting systems (i.e. when they are applied in real elections in a real society). It is therefore necessary to continue the study about society characteristics for creating more realistic environment models.

References

1. D. Jefferson, A.D. Rubin, B. Simons, D. Wagner. A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE). 2004.
2. A Ansper, A. Buldas, M. Oruaas, J. Priisalu, A. Veldre, J. Willemson, K. Kivinurm. The Security of Conception of E-voting: Analysis and Measures. E-hääletamise kontseptsiooni turve: analüüs ja meetmed. 2003.
3. A. Buldas, P. Laud, J. Priisalu, M. Saarepera, J. Willemson. Rational Choice of Security Measures via Multi-Parameter Attack Trees. In *Critical Information Infrasturctures Security Workshop - CRITIS 2006*, LNCS 4347, pp.235-248, 2006.
4. Estonian National Electoral Committee home page, `http://www.vvk.ee`
5. B. Schneier. Attack Trees. Dr. Dobb's Journal December 1999. Modeling Security Threats.
6. D. Geer, K. Soo Hoo, A. Jaquith. Information Security: Why the Future Belongs to the Quants? *IEEE Security and Privacy.* July/Aug, 2003. pp.32-40.
7. Interview with Estonian public prosecutor Mr. Margus Kurm.
8. M. Surf, A. Shulman. How safe is it out there? Zeroing in on the vulnerabilities of application security. Imperva Application Defense Center. 2004.
9. The elections' atlas of the USA. `http://www.uselectionatlas.org/`
10. The Parlimental Elections in Estonia in 2003. `http://www.vvk.ee/r03/yld_kulud.stm`
11. Department of Defense Washington Headquarters Services Federal Voting assistance Program. Voting Over the Internet Pilot Project Assessment Report. 2001.
12. E-voting System: Overview. Estonian National Electoral Committee. 2005. `http://www.vvk.ee/elektr/docs/Yldkirjeldus-eng.pdf`
13. Triinu Mgi. Practical Security Analysis of E-voting Systems. Msc Thesis. Tallinn University of Technology, 2007. `http://triinu.net/pro/`