National Electoral Committee

# E-voting concept security: analysis and measures

Authors:
Arne Ansper
Ahto Buldas
Aivo Jürgenson
Mart Oruaas
Jaan Priisalu
Kaido Raiend
Anto Veldre
Jan Willemson
Kaur Virunurm

Document: EH-02-02
Date: 27 December 2010

- Tallinn 2010 –

# - Contents

# 1. INTRODUCTION

This analysis has been commissioned by the National Electoral Committee with the aim of tackling the issue of security risks in the technical conception of e-voting used by the National Electoral Committee. The task is to appraise the general suitability of security in the proposed solution, to map the technical and organisational risks as thoroughly as possible, and to amend the security requirements presented by the system if necessary.

The first version of the document was completed in 2003, when the risks of the e-voting solution, that was then still in the planning stage, were analysed. In seven years e-voting has become an everyday reality and during this period the technical solution corresponding to the original concept has been used altogether in five e-votings. Practice shows that safe conducting of elections is possible and the original concept has withstood the test of time. In 2010 the document was updated on the basis of the experience acquired during e-votings that had taken place in the meantime and new information about the security risks of IT and the Internet.

Taking into account the relative simplicity of the analysed voting scheme, pinpointing the main risks is an intuitive process. The concept itself includes both a security analysis and protection measures against the major risks – digital signature, encrypting the vote, the division of Central System into several servers. The present analysis is more systematic, it discusses risks in greater detail and sets many more specific technical requirements.

Our work is confined to technical security and work organisation processes. We do not assess political risks or evaluate social or political aspects of electronic voting. Nevertheless, we do highlight issues related to security that are essentially technical but require a decision or acceptance at a higher, political level. Such aspects include the conflict between privacy and controllability of the voting process, risks related to the centralisation of voting, dangers connected with reliability, and the necessity of technically competent auditing.

# 2. REQUIREMENTS AND PREREQUISITES OF THE E-VOTING PROCESS

## *2.1. The problem of conflicting requirements*

The conflict between controllability and confidentiality makes secret voting complicated. On the one hand, the correctness of results has to be guaranteed, therefore the whole process has to be auditable from the beginning to the end, a trace must remain of every activity. On the other hand, the confidentiality of votes has to be guaranteed in order to preserve the democratic nature of the process; thus there should be no possibility of ever establishing a link between the voter and the vote at any point of the process. These two requirements – controllability and confidentiality – are contradictory in essence. Further requirements – the necessity to control the right of the voter to choose, the prohibition of repeated voting, diversity of voting methods etc. give rise to further problems.

There is no ideal solution, a compromise will have to be reached.

The compromise involved in conventional voting procedures is the use of multiple envelopes and a number of complicated checking procedures related to them. The number of risks involved is high – the necessity to trust polling division staff, the impossibility to provide complete satisfaction to protests etc. However, these are accepted with the hope that the choice of the society will be reflected in the process with sufficient, though not absolute precision.

In terms of e-voting as an IT task this implies that *requirements* to be set for the system have to be agreed on at a political level and consciously. Auditability and secrecy, error protection and unprovability, security and comfort do not go hand in hand; a line should be drawn somewhere and a decision should be made. Nearly all e-voting security studies discuss this contradiction inherent in the task. The best analysis made so far is apparently that produced by Peter Neumann [Neumann].

## *2.2. The set requirements*

A systematic description of system security requirements along with clarifications is presented below. Most of them are directly described in the e-voting concept, some originate from our Constitution and election acts and others are simply "classical" security requirements set for e-voting systems.

### 2.2.1. Correctness of voting requirement

Correctness or integrity of voting includes functional requirements the fulfilment of which guarantees the result of the voting is correct, reflects the choice of voters and is in conformity with the law. There is a whole number of such requirements – in essence, the whole text of acts regulating elections is a list of such requirements – and the whole design of the voting scheme is concerned with meeting these requirements. The following is a list of only the most important security requirements.

*Authorisation of voters* – only voters whose names are included in the voters' list can vote, and one can only vote for the candidates of one's electoral district. The authorisation requirement in its turn entails the necessity to *authenticate* the voter.
*"One person – one vote" –* of all votes given by the voter, only one vote has to be considered, regardless of the way the votes were given.
*Prohibition of falsification of votes* – no-one should be able to change votes given by voters or add falsified votes to the system (e.g. vote in place of voters who did not participate in the elections).
*Uniformity of voting* – equal voting possibilities should be ensured to all voters.
*Possibility for electronic re-vote* – the voter should have the possibility to re-vote.
*Supremacy of conventional voting* – any other method of voting annuls all the e-votes given by the voter.
The following are two functional requirements that are often set to e-voting systems, but not accepted in the Estonian law and not directly supported by our technical solution of e-voting.
*Annullability of votes by the voter –* the possibility to annul one's already given vote.
*Possibility to give an empty vote* – possibility to vote "for no-one" or to give an empty vote.
The function of giving empty votes has been introduced for two reasons – a technical and a political one.
The technical reason is to enable users who do not wish to vote to ensure for their peace of mind that no-one else uses their name for voting. The political reason is to give citizens the possibility to express "democratic protest" by means of a demonstrative non-use of their citizens' rights.
In fact both requirements are realised on the basis of the supremacy of conventional voting. The voter can after e-voting always go in person to the polling station and deposit an empty ballot paper in the ballot box there.

### 2.2.2. Secrecy of voting requirement

*Secrecy of vote* – no-one should at any point find out who the voter has voted for.
*Privacy of the fact of voting* – it should be impossible to identify if the voter voted, the time of voting and the computer the vote was given from.
The fact of voting is never completely secret neither in case of conventional nor electronic voting. Internet service provider (ISP) can trace its clients' connections to the web server of the National Electoral Committee; an observer watching the outer door of the polling division can see the people entering the division. At the same time none of them can ever tell whether the voter has actually voted. The same sort of "light" protection is needed for e-voting.
*Unprovability of voting* – the voter should not be able to prove for whom, when and in what way he/she voted.
Unprovability is a method aimed at protecting *voluntarity* (freedom of voting, uncoercibility). Uncoercibility requires that the voters are free in their choice. The impossibility for the voter to prove how he/she voted rules out controllable selling and buying of votes or other forms of coercion (e.g. employer's pressure).

*Secrecy of voting result –* the results of e-voting should not be known to anyone before the end of the conventional voting; the electoral committee will not disclose the division of e-votes separately from general voting results.

#### 2.2.3. Operating reliability of voting requirement

*Operability of the voting system* – the technical system of e-voting has to be reliable, available to voters and those responsible for the organisation of voting, operate with adequate speed, ensure the preservation of data and timely presentation of voting results.

#### 2.2.4. Reliability of voting requirement

The society and the parties involved have to believe both before and after the voting that e-voting is (and was) a trustworthy way of giving one's vote. In technical terms this involves the following requirements.
*Transparency* – the process and mechanisms of voting have to be public and understandable.
*Auditability* – specifically authorised persons must have the opportunity to be convinced that the whole process of voting has been conducted correctly.
*Controllability of vote counting* – every voter should have the possibility, should he or she require so, to check whether his or her vote has been taken into account in the counting of votes.
*Repeatability of counting* – the process of counting e-votes has to be repeatable.

#### 2.2.5. Theoretical requirements

In the interest of integrity we shall present two additional requirements set for the records of voting. These are *universal verifiability*, in the case of which every interested person (including persons not engaged in the system) should be able to prove the final calculation of results, and *absolute (fail-safe) of votes*, or in other words the requirement that under no circumstances should a voter's vote become public, including the situation where there is a conspiracy between all the other parties (e.g. those responsible for the organisation of voting).
We believe that there is no real voting scheme meeting all these requirements, and will never be.

### *2.3. Technical prerequisites*

Technical prerequisites on which the presented conception and the current analysis are based are as follows:
*Central System servers* are secure and reliable. That means that compromising a Central System server may affect the security of e-voting to such extent that the results will have to be annulled.
At the same time the Central System is still divided between several servers. Is this necessary in the case of the abovementioned prerequisite? The answer is definitely yes; such modularity enables to considerably improve the technical security of the system.

The *Central System* along with the *intranet* is integral and operable and its physical security is in order. The analysis does not include intermediation attacks or connection and power interruptions taking place in the intranet.
The information system of e-voting is separated from the rest of the voting information system *at the network level*, the connection with the outside world takes place by means of limited interfaces.
The voter has an instrument for strong authentication and digital signing, like an ID card, DigiID or Mobile-ID with valid *authentication* or *digital signature certificate*.
*Digital signature* is unfalsifiable, ID card, Mobile-ID and the basic software related to their use are secure and error-free.
*Voter's environment (computer, browser)* is secure. At the same time this environment is uncontrollable for the electoral committee. The owner of the environment is responsible for its security and the risks connected with it should be managed by informing the citizens and raising information security awareness.
*Input data* of the system – lists of candidates and voters – are correct.

## *2.4. Architectural components of the system*

The current analysis is not concerned with describing the system itself insofar as this task is covered in the analysed concept. However, in order to reach common understanding, names should be attributed to components of the system and data processed in it. The list is not complete – most of the definitions are either intuitive or they have been described in the analysed concept.
*Voter's application, VA* – application which encrypts the vote in the voter's computer and gives signature. Voter's application operates in the *voter's computer*
*Vote-transfer server, VTS* – server which supplies voters with the application and supporting data, receives given votes and transfers them to the VSS. Since VTS has also been used as the web site of the National Electoral Committee defined in the law, it is often referred to as the *Web server*.
*Vote storage server, VSS* – server which stores the encrypted e-votes given by the voters and enables to sort, delete and forward them to the VCA. When e-votes are forwarded to the VCA, the voter's digital signature is separated from the encrypted vote and the e-votes become anonymous.
*Vote counting application, VCA* – a separate application which un-encrypts digitally unsigned e-votes, sums them up and delivers the results of the e-voting. The computer running the VCA is called *VCA server.*
*Validity confirmation service* – an external service confirming the time when digital signatures of the votes were given and the validity of the voter's signature certificate.

*Internet* – network connection between the VA and the Central System.
*Intranet* – connections between components of the Central System. The intranet also includes the *firewall* and other possible mechanisms of access control at the network level along with offline data carriers between VSS and VCA.
*Audit system* – component of the Central System dealing with gathering audit data and working with *audit application*.

Besides that, the system contains a *Database* and a number of *applications* – vote sorting application, audit application, voters' feedback application etc. The database is located on the VSS, although it is an independent logical component and could be placed on a separate server if necessary.
The *elections information system* which generates data required for the e-voting, where annulments and restorations are received from and where e-voting results confirmed by the electoral committee are recorded also indirectly belongs to the e-voting system.

# 3. IDENTIFIED RISKS

A detailed risk analysis based on the architecture of the system and categories of risk is presented in Annex 3. This chapter deals with conceptual problems related to e-voting and sets out consolidate results of the technical analysis.

## *3.1. Fundamental problems*

### 3.1.1. Necessity to trust the voter's computer

The complexity of a modern personal computer has reached such a level that, from the voter's point of view, it is a "black box" nobody can or is able to control. The computer can do virtually everything on behalf of the voter but behind his/her back – vote for another candidate, sign some other document in addition to the vote, send the voter's vote openly to the press and so on.

In general four areas of the voter's computer can be exposed to attacks:

– network / operating system (for example, Microsoft Windows network library errors, risks connected with using USB memory devices, etc.),

– e-mail, instant messaging programmes and social networks as the most widely used internet services,

– web browser security errors (every browser and its extensions contain sufficient amount of errors to enable attacking a computer during visiting web sites containing attack codes),

– physical access.

Using these channels, one can install software which may:

– trace user's actions – gets to know his vote and/or the PIN code of his or her ID card, or

– replace the voter's application with a different one and give a wrong (different from that of the user) vote, or

– misuse the ID card and give digital signatures of the voter who is not aware of the fact, or

– block the vote.

Such risks existed in 2003, when the first version of this analysis was compiled, and they still existed in 2010. In that aspect, the situation has not changed. Attacks are made more often, attacks may be better prepared and most attacks have criminal purposes.

By today we know about one attack in a foreign state where a chip card connected with a computer was used and financial transactions were made in the name of the user so that the user did not notice it. The way of attacking that was known before and considered technically possible has now been demonstrated in practice, but under circumstances that do yet not significantly change the risk assessment of e-voting.

In Estonia, a theoretically interesting situation has arisen by today, where two different versions of ID card drivers have been created; one of them has an open source code that is accessible to all. Because of the abundance of basic software, the end users have to make a choice between them, and the unsuspecting computer user often is not able to tell which ID card drivers are safe and which are not.
At the same time discovering of attacks, treating them as crimes, investigating them and finding the criminals has developed significantly both in Estonia and the rest of the world. More attention is paid to protecting computers and in conclusion the situation in practice has not changed much for the end user in regard to risks.

In greater detail the problems related to the computer of the voter are discussed in Avi Rubin's work [Rubin], and the specific risks connected with the system of Estonian e-voting are also dealt with in the article by Buldas and Mägi [BM]. Unfortunately these are risks that the Central System of e-voting can neither control nor avoid.
But still we find that trusting the voter's computer in spite of the known attacks against it is an acceptable risk in carrying out the voting.
The reasons for that are as follows:

– There is no sense in attacking one voter's computer or network connection in order to falsify his or her vote. Only a massive attack can have an effect on the whole result of the voting.

– Attacking a voter's computer with the aim of acquiring his/her passwords/PIN codes is not related to the event of e-voting, this can equally be done at any other time.

– Attacking a voter's computer, assuming that this computer is used by one or another concrete person, is not practical today. Most attacks are automated and meant for mass use.

– A secret massive attack on computers is practically impossible. This has been demonstrated by the way malware has been spreading – even the most artful and stealthy malware exposes itself in some kind of computers, if for no other reason, then because of computer's own errors (those of Windows, browser, Outlook, etc.). Even the most successful malware of recent times, the Stuxnet that was created to spread itself very unnoticeably and expressed its influence in only a few situations, was discovered in spite of that. Differences in configurations of voters' computers along with the diversity of operation systems, browsers, antivirus software, etc. create the situation where some of the voters will notice the attack and it will be discovered and blocked.

The act of e-voting itself may fail because of such an attack, but most probably the attacker and the extent of the attack will be identified. It is clear that the success of the attack will remain limited and the original aim of the attacker will not be realised. Therefore commissioning such an attack as a political order is not realistic.

– It is possible to conduct a massive falsification attack on a web server if there is no active communication with voters from the web server's address.

– Given the fact that voting takes place by using non-standard technology in web terms (a separate application is used), the falsification of votes requires considerable knowledge. It is much easier to attack an internet-bank and the profit gained from it is considerably more realistic.

### 3.1.2. Need to trust public Internet access points

Computers used by many people should be considered separately: Internet access points and cyber cafés, school computer classes, centrally managed computers of large companies and also the so-called voting tents that offer the possibility of e-voting and are erected especially for the elections.
Administrators of such computers and networks have the possibility to attack all voters using these computers, for instance by allowing them to vote only for the political party of their preference. School computer class can of course be compromised by a simply clever schoolchild.

There have been cases of similar attacks on internet banks. Yet, as far as it is known these have only happened a few times. We still think that this risk, too, is small and cannot influence a great number of e-voters.
Instead, we call attention to a significantly greater risk that is not of technical nature – the risk that an Internet access point (IAP) administrator has a greater than average possibility to direct or influence the decisions of voters on the spot. In this context, people should be warned to avoid using IAPs (especially those belonging to a political party) if possible.

### 3.1.3. Need to trust public network

In addition to his or her computer the network user has to trust public Internet in all its complexity.
The voter ***has to*** start voting on the right web page. It is the basis of security of the whole voting process. If someone succeeds in making the voter begin voting from a wrong page, it will not be possible to establish limits to further actions: he/she can be sent a wrong application, given a wrong list of candidates and his/her whole computer can be taken under control. These risks are analysed in greater detail in the risk analysis chapter 9.2.2 "Risks related to Internet usage".

### 3.1.4. Need to trust Central System computers

The system layer of the Central System – operation systems and standard software or the "black boxes" described in the concept – consists of components that we simply have to trust. This need to trust can be reduced to a minimum simply by procuring those components from reliable sources who should thereby not be aware how it is planned to use the components. However, the fundamental problem remains to be solved.

### 3.1.5. Impossibility to support all voters

The use of any kind of technical equipment leads to the exclusion of a certain part of people who cannot use such technologies because they do not have access to computers, do not have the necessary computer skills, a physical or mental disability prevents them from using a computer, a person has no wish to use a computer or a person just uses a platform that is not widespread. Voter's application will certainly not work on all computers citizens of the Republic of Estonia have access to. Although the application will be created for Windows, Linux and MacOSX platforms, a certain number of platforms and operation systems will remain unsupported. Moreover, there is no official standard introduced in Estonia which would establish the legally "permissible" computer platforms the e-voting would be able to support.

### 3.1.6. Possible conflicts of conventional and e-voting processes

E-voting takes place at the same time as conventional voting; most probably the same persons will conduct the voting and are responsible for it.
This means that conflicts may arise between the processes of conventional voting and e-voting. People will have to do several things at a time, know more things than before and divide their attention between several systems.
The priority conflict of polling divisions can be given as an example. In order to start counting e-votes, *all* annulment applications must have been received from *all* the polling divisions, they have to be transferred by electoral district committees to the National Electoral Committee, and from there they must have been sent to the Central System. A delay in the work of even one polling division blocks the whole process of completing e-voting. Nevertheless, compiling and sending these lists is not a particularly important job for the polling divisions insofar as the result of voting of the polling division does not depend on it.

### 3.1.7. Risks related to the centralisation of processes

The conception lays out a centralised voting and counting scheme.
Centralisation improves efficiency. Yet, it leads to a concentration of risks. This concentration concerns both the human and technical level – a single programmer's error or dishonesty of a person counting votes can have a considerably greater effect than that of conventional voting based on detached process.

### 3.1.8. Risks arising from formalisation of processes

The rules of physical world are always "soft" because relations between human beings can always be changed. In information systems, however, the rules are rigid and it is not possible to ignore them or "cut corners".
As a result, overformalised procedures can completely block the work or lead to the disappearance of essentially reasonable ways of making exceptions that have worked so far.
Identifying a voter can be given as an example here. In conventional elections a father of a family who has come to vote with his wife and children can vote even if he has left his passport at home; the people at the polling division trust the words of his wife, check an employment certificate or a sports club membership card and take the risk of

false authentication. There is no such possibility in e-voting; the vote cannot be given without an ID card insofar as VA ↔ VSS protocol contains no "begging" messages.
A more drastic example is the realisation of London ambulance dispatch system by using IT means. The previously informal management of ambulance transport was terminated and as a consequence 26 people died in three days because the help did not reach them.
It is also necessary to consider these procedures carried out at the elections where tendencies exist at present to break the rules, and think whether problems can arise in similar e-voting procedures. Here it is necessary to get an input from as low level of the organisation of conventional elections as possible.

### 3.1.9. Unauthorised changing of input and output data of the system

E-voting system can be viewed as a mechanism which receives the input of candidates' and voters' data and voters' choice, and produces the output of the voting results and verification data confirming it.
Correctness of all these input and output components is of critical importance. There is no sense in having technical security if one does not check whether the numbers inserted into the voting system of the National Electoral Committee correspond with those produced by the VCA.

### 3.1.10. Development and management problems

Two highest risks of every information system are the *quality of development* or *software faults* and the *quality of management* or *system configuration faults*. Both untested software and negligent management can cause errors and trigger security problems. The e-voting system is particularly exposed to such errors since it is a dispersed system (consisting of components functioning in several different environments) that is rarely used, difficult to test and with a time-critical deadline.
*Application quality problems arising from the use of cryptography*
The voting scheme uses the public key infrastructure (PKI) both for the servers and the VA, the operations that are carried out are thus relatively simple. Practice shows, however, that because of the complexity of details the realisation of the PKI involves a lot of mistakes whereby a tiny error can lead to complete failure of security. For example, the development of Microsoft's PKI involved very serious errors (signature chain safeguard failure).
In terms of e-voting, cryptography-related problems are first of all to be expected in the operating reliability of the VA (uncontrollable environment) and during the counting of votes (complex key management).
There is no simple remedy. Correctness of application has to be ensured through thorough analysis and testing. More time and money is required for developing PKI applications than for software having similar functional complexity but not relying on cryptography. Although developing and using mission-critical software is very expensive and complicated, it is being successfully done in the fields of space research, energy, etc. The experience in realising the Estonian e-voting solution has shown that we, too, can do that.

## *3.2. Technical risk analysis summary*

The most important technical risks can be divided roughly into four categories:

– Risks deriving from the Internet as an open and public environment;

– Errors happening during voting that are aggravated by using an application that is unknown to the voter;

– Errors of vote recording/sorting server as the most complex component of the system;

– Vote counting problems, magnified by high requirements to the organisational security of the process.

A detailed analysis is presented in chapter "Annex 3 – Technical Risk Analysis", the following is a list of what we consider the ten greatest risks (with the greatest impact and probability).

| **Risk** | **Location** |
|---|---|
| Attack against user's computer and gaining control over it | OS, applications |
| Failures and quality problems of voter application | VA |
| Man-in-the-middle attacks against web server and voter's computer, fake web pages | Internet |
| Compromising of voter application or its input data | VA, VFS |
| Defacement of web server or unauthorised changing of its contents | VFS |
| Violation of vote confidentiality during voting in web server | VFS |
| Traditional web application/ web server management and security errors | VFS |
| Failures and quality problems of Central System (VSS) software | VSS |
| Functional failures of VCA | VCA |
| Destruction/ inaccessibility of VCA secret key | Key management |

The highest security risk is the security of web server contents and applications. Here, the great probability of risks and easy execution of attacks combine with the effect influencing the whole voting process.

We did not identify any formerly unacknowledged fundamental problems that have not been taken into account in the concept. The simplicity of the scheme means that errors are also simple and the intuitive result is not much worse than technically systematic risk analysis.

In conclusion it can be said that the risks of e-voting are in fact very similar to the risks of conventional voting, most technical attacks and threats have analogies in the real world. Instead of IT-systems, there are people and organisations, but schemes and processes are the same. Instead of voter application errors election lists could be wrong/faulty ("Florida butterfly ballot"), software and people alike make mistakes in checking and counting the votes, problems arise with operability (queues form at polling stations) and reliability is attacked (protests).

E-voting only adds dependency on technical equipment. Besides that, general effect of problem magnification arises from the use of technical equipment: frequency of errors is reduced but their scope becomes larger.

■

# 4. REQUIRED AND RECOMMENDED SECURITY MEASURES

We will not give room here to standard requirements to secure system construction – it is possible to read about them in many books or standards on data security. We presume that access control is applied, management activities are documented, operating systems (the "black boxes" of the concept) updated in the field of security errors, applications check their input and log the activities carried out, and so on.

**We recommend that the general security of the system is set on protection level High (H) of the three-level IT baseline protection system (**Estonian: *Infosüsteemide Kolmeastmeline Etalonturbe Süsteem, ISKE***).**

At the same time the security of a system of vital importance cannot be limited to baseline protection alone. We will therefore list here the security requirements deriving from specific features of the e-voting system.

## *4.1. General requirements to the Central System*

### 4.1.1. Requirements to Central System architecture

Selection principles of operating systems and data base of the Central System are described in the concept and have to be respected. The objective is simplicity and checkability.

**Separation of the Central System**

E-voting Central System must be a separate autonomous information system with autonomous servers and network connection between them.

**Network zoning**

Web server/VFS must be located in a separate part of network, but not between the firewall and the public network.

**Restricting the functionality of systematic platforms**

The functionality of all servers and other parts of the system must be the absolute minimum necessary for providing required services and running applications. Servers should not contain development means (compilers, support of superfluous programming languages), data base access means, etc. Unnecessary applications and services must not be installed, the firewalls may contain only a permitted minimum number of absolutely necessary network protocols, etc.

**Detection of network level attacks and ensuring system integrity**

The system must detect in real time the network level attacks (intrusion detection system (IDS) at network level) committed against it. Servers must detect violation of the integrity of files important for the operating system and e-voting. Some Tripwire-type mechanism may be used for that.

**Recording of servers' status**

Before the beginning of the e-voting, so-called "clean" copies must be made of the Central System servers, containing the whole configuration and software of the servers. After this, it is possible to restore the server which failed during the voting into functioning configuration as soon as possible.

After the end of the voting period, a "frozen" auditing copy must be made of the server hard disks of all VFS and VSS servers and the data contained in them.
A second auditing copy must be created just before the final establishing of the results.
Auditing copies must be stored securely – in a security envelope and locked safe – and the all access to them must be recorded.

**Use of central data base**

There must be one common data base of candidates and voters for all system components and it should be located in VSS (or in a separate data base server). VCA constitutes an obvious exception because it requires all data to be submitted as static files.

#### 4.1.2. Requirements to Central System applications

**Non-graphicity of user interfaces**

Since all operation carried out are extremely simple, we recommend that only applications with textual or pseudo-graphical interface be used in the whole Central System. This enables the use of simpler development instruments, improvement of application transparency, not having to install graphical interfaces in the Central System servers, etc. The negative side is the more Spartan (more technical, less attractive) appearance of the applications. Besides that, it will be necessary to document their use must in detail, but this is a virtue, not a downfall.

**Logging of technical errors**

All Central System applications must register and forward to the auditing system the technical errors and logical controversies that occur during their work.

#### 4.1.3. Ensuring reliability

Here we will describe the architectural and technical measures for ensuring reliability. We must not forget the postulate of risk analysis – reliability is the most threatened by management errors and faulty software.

**Specification of system operability requirements**

System operability requirements must be specified. It must be known how quickly the voter must get a reply, how much time is allowed for sorting and counting of votes, etc. These constitute input data for the technical design of the system.

**Load tests**

The system must pass a load test and a stress test.

**Monitoring**

There must be an application monitoring the work of the system as a whole, that would save all collected capacity data. In case of errors notification should be sent to the system manager.

**Restriction of data loss, data restorability**

The amount of data that might be lost in case of Central System errors must be limited. There are two methods for this – duplication and the possibility of repetition of all input data, or mirroring of data to another system with the help of software.

The simpler method is to make frequent back-up copies of data generated during e-voting (of data base redo-logs). As the volume of data is small, the frequency of saving back-up copies could be set conveniently high (e.g. every five minutes).

**Recovery plan**

Refreshment procedures of the whole system or data base must be in place for the situation where a component or data base has been destroyed for some reason (hardware failure, management error, etc.).

#### 4.1.4. Requirements to data format

Data formats must be as simple as possible.
Human-readable formats are preferred.
XML is not a preference in itself, except for external channels (annulments/restorations).

**Ensuring integrity of data transported**

When transporting data between VFS – VSS and VSS – VCA, measures should be implemented for ensuring the integrity of data transferred. Integrity must be checked during data transport as well as later, during auditing.
Checks required during auditing are described in the relevant section.
Simpler measures should be implemented during transport, such as calculating checksums.

**VCA input and output to plain text**

The whole VCA input and output must be in plain text (e.g. CSV format).
XML (or any other SGML-based format) would make VCA too complicated.

**Falsification-proof logging in plain text**

Falsification-proof logging should be used for logs under auditing.
Logs should be readable in plain text.

**Format of vote cryptogram**

Open form vote should be in the simplest format possible – preferably ASCII text.
We recommend that standard PKCS#1 2.1 encrypting scheme RSAES-OAEP be used for encrypting votes and default functions (PKCS) included in the standard be used as support functions. In practice this means that votes are encrypted directly using RSA algorithm, without interim symmetric encrypting. Although this sets limits to the length of the vote and does not suit complex voting schemes (multi-choice, with room for remarks), it is the best choice for the Estonian scheme.
A vote can be signed with any digital signature certificate which does not have an application field restriction on e-voting. At the moment, ID card, DigiID and Mobile-ID can be used for that. In the future there may be other digital signature certificates that can be used then.

#### 4.1.5. Requirements to external data channels

**Accessibility of candidate lists**

Everyone who wishes must have the possibility to get a copy of the complete candidate list.
NEC must publish the checksum of this list via an independent channel.

**Input data integrity check**

The polling list and the candidate list in the system must be accessible for comparison with the originals of AS Andmevara and NEC.
Thus, there could be a request in the Central System and NEC data bases which calculates the checksum over the personal identification codes of the voters entered in the list of voters.

**Output data integrity check**

Files leaving the system (voting results, lists of e-voters) must be comparable in some way to the data in the system.

**Signing of annulment and restoration lists**

Annulment and confirmation lists sent out by NEC must be digitally signed. VSS annulment/restoration application must check the signature against the signature of authorised signatories in VSS, which must include at least two persons.

## *4.2. Requirements for the components of the system*

### 4.2.1. Requirements for VA

Official **authentication certificate** on the authentication instrument, and no other certificates that may also be on the authentication instrument, should be used to identify the voter.
The application must not buffer the access codes of the voter's chip card certificates and must ensure, whenever possible, that also the operation system libraries and other basic software would not allow buffering.

**Hiding the voter's choice and the data viewed from the web server**

Viewing the data on candidates in the browser / VA must be totally independent from the web server. All information necessary for casting the vote must be sent to the browser in one enquiry, so that the web server would have no knowledge which candidates' data the voter viewed.

### 4.2.2. Requirements for VFS / Web server

**Authentication of the web server by the voter, HTTPS**

Communication between the web server and the voter's computer / VA ***must*** be secure.
Authentication of the server is primary, encryption of the channel is secondary.
It is the most important requirement to the voting process. If the voter goes to wrong web server, it is the same as beginning to vote in a party headquarters instead of a polling division: nothing can be guaranteed, the result may have no connection with the will of the voter.
The server must work in a secured system, i.e. use HTPPS protocol.
The voter can check the authenticity of the web server through its *certificate*.
The certificate *does not have to be* signed by the certification server the voter's computer trusts; real security is created when the voter checks the checksum ("fingerprint") of the server's certificate. This possibility exists in all Internet browsers.

During the informing process the voters should be informed how to check the server's certificate and what is the correct checksum, and requested (or at least strongly advised) to check it.

**Minimal functionality**

As the server is in public internet, and can be attacked through all applications working there and through all open services/protocols, only such components that are necessary may be present (not only working, but also be installed!) in the web server.

**Authentication of the voter with authentication certificate of authentication instrument**

The voter must be authenticated with the authentication certificate and in no other way. It should also be checked that the given vote is digitally signed by the same person that authenticated himself or herself to the web server.

**The only function of the HTTP web server should be redirection of HTTP**

For the convenience of the voters, it may still be decided to keep up the HTTP service, too.

In that case the only function of the server should be redirection of HTTP to the actual safe HTTPS web page.

**The domain used by e-voting must be located in .ee top-level domain.**

We recommend reserving a separate server name (FQDN) for e-voting, that is used only for e-voting.

**Restricting the data on candidates displayed by the web server**

Only such information on the candidate that has been officially deemed necessary and is uniform for all may be displayed on e-voting page or VA.

There should be no references to advertising materials, like the candidates' home pages.

**The contents of the web server must be as static as possible**

The web pages for showing the data on candidates, loading VA and giving help must not be in the data base. If it is necessary to keep the data in the data base because of the large number of candidates or some other reason, a static copy should be generated for the web server.

**Static, standard, validated HTML**

The web pages displayed to the voter should be written in static HTML that does not use active scripts implemented in the browser or server.

Web pages should be validated following the simplest (minimalist) HTML standard possible.

**Logging of correct votes**

VFS must log the correct votes forwarded to VSS (Log1).

This is essentially the only possibility to audit the work of VSS.

**Not logging of faulty votes**

When VFS establishes that the e-vote received from the voter is technically faulty, such votes must not be saved. It is possible that the fault is in encrypting and saving would violate the confidentiality of the vote. This is a problem in spite of the fact that this concrete vote was not taken into account in voting.

In conventional security this request corresponds to the ban of logging on failed authentication attempts data ("wrong password").

However, the fact of receiving faulty vote should be logged.

**Logging of technical errors**

VFS must log all faults in the voting process. Discrepancies between authenticated person and signer of the vote, interrupted sessions (for example, confirmation of the acceptance of the vote is not sent to the voter), etc. must be registered.

**Use of reverse proxy**

All the replies sent by the web server to the voter must be routed through reverse proxy which will carry out elementary security control over their content.

**Multitasking of VFS application**

VFS must be able to serve several voters simultaneously.

It must be taken into account that forwarding a vote to VSS and waiting for a reply from VSS may take time.

**Creating VFS–VSS connection**

Permanent channels must be created for data exchange between VFS and VSS.

Both their lowest and highest number must be restricted. Restricting the highest number of connections will prevent overloading VSS by VFS (risk of sending a vote ten times). Essentially it means that denial-of-service attacks at application level will not get further than VFS.

The channels must be created by VSS, i.e. connections will be from the intranet to the outside.

**Confirmation of acceptance/rejection of vote**

VFS must send to the VA or browser confirmation from VSS about accepting or rejecting the received vote. Confirmation must be final, the vote that has received positive confirmation must be really saved in VSS. Naturally this does not preclude later annulment of the vote for the reasons shown in the concept (taking part in conventional voting etc.).

**Identification of attacks at application level**

It is necessary to monitor or at least to analyse the faults and attacks emerging at application level of voting.

This is not easy because it requires the prediction and recognition of possible attack patterns. In the case of conventional systems the logs of daily use give a possibility for that, in e-voting there is no such data. If monitoring is not possible, this analysis must be made later, with the help of audit system.

Model rules for monitoring are the following:

- more than N authentications for one voter;
- more than N votes given by one voter;
- more than N authentications for one voter, that are not followed by giving the vote;
- more than N authentications in a very short period of time;
- discrepancy between authenticated voter and signer of the vote;
- giving technically faulty (wrong format, unsigned, ...) votes, etc.

#### 4.2.3. Requirements for VSS

VSS is the most complicated component of the e-voting system.

Most of necessary administrative activities, like sorting of votes, creating lists of e-voters, entering annulment lists and restorations etc., are carried out through VSS.

**Security of VSS applications**
The most elemental and the most important requirement is the "classical" security of applications working in VSS.
It makes no sense to give a list here – it is important that VSS applications work according to specification and follow the ordinary security rules. Users must be authenticated, there must be a log on each activity, passwords (if they are used) must be kept and forwarded encrypted, applications and users must not have more rights than necessary, and so on.
**Correctness of VSS applications**
If errors are found in VSS sorting applications during voting, it will probably bring along the need for extraordinary direct access to e-votes data. Each such access is a possible violation of security requirements and procedures. Therefore the thorough testing of the efficiency of VSS applications is also a security measure.
**Restricting the rights of VSS applications and users**
The applications should neither have nor give to the users more rights than necessary – for example, the vote sorting process must not be able to change the list of candidates.
It can be realised through operation system access controls, data base design and giving minimal necessary data base rights to different applications.
**The state of data base at each moment of time must be retrospectively identifiable**
For that, all items induced to the data base must have a time stamp, when changes are made, the preceding state of the item must be archived, etc.
**Freezing the parameter tables and constant data of data base**
VSS data base contains data that must not be changed during e-voting: lists of candidates and electoral districts, parameter tables, information on votes given, etc. Data base rights must guarantee that they really ***cannot be*** changed.
As the state of some parts of data base is finally fixed only at the end of the voting period, such tables must be marked non-writeable only then.
**Additional controls of VSS applications**
VSS applications influence the result of voting so directly that in order to avoid possible mistakes, several controls should be imposed.
For example, the vote annulment procedure (movement of conventional and digitally changed data) is described in the concept. The requirement of programmed follow-up control should also be added: after "feeding" the annulment list into the system, it must be checked if the VSS application is planning to cancel as many votes (and also randomly – the same votes) as there are in NEC annulment list.

#### 4.2.4. Requirements for VCA and VCA server

**Separation of VCA server from Central System network**
VCA must not have network connection. All communication with the outside world may take place only via removable media (CD, floppy disk, USB memory, printer paper).
**VCA input and output to plain text**

The whole VCA input and output, incl. candidate list, must be in plain text (e.g. CSV format). XML or any other SGML-based format would make VCA too complicated.

**Requirements for protecting VCA memory**

Data processing must take place in main memory. VCA must neither display nor save intermediate results (votes counted, state of the moment) – all data processing must take place only in the memory of the application.

After counting and export of votes the VCA server should be booted, and the server should be kept disconnected from the power network for at least three minutes.

**Vote format controls**

Before all other controls, type and format control must be carried out on decoded vote. Logical controls described in the concept (if the candidate with such number exists, etc.) can be carried out only after that.

The reason for that is the fact that the vote is direct and uncontrolled data channel from the outside world to VCA. Before VCA nobody can look into the encrypted vote, or see whether the cryptogram really contains the number of the candidate or program code to be invoked.

**Printout of voting results directly from VCA**

We recommend that immediately after counting the votes the voting result should be directly printed out from VCA and signed by all members of the committee. This is the so-called "original" of the voting result.

#### 4.2.5. Requirements for auditing system

**Data collected by the auditing system**

Auditing system must collect:

- functional logs – LogWeb, Log1 ... Log5;
- technical logs of VFS, VSS and VCA applications;
- IDS logs, tripwire logs;
- console logs of system servers;
- logs of the log-ins of the users of system servers (utmp), etc.;
- error reports in free text (if they arise; to be entered manually);
- records of activities (key management, vote counting).

Auditing system must be secured on the same level as other Central System components.

#### 4.2.6. Requirements for the management process of the system

Measures described in reference security deal with the requirements for management system fairly thoroughly. We just repeat the need to **document** management activities and **save the console logs** of all servers of the system.

For each new e-voting all Central System servers must be re-installed and configured. For starting, the operation system must be clean and as new as possible; all other software, too, must be installed and configured, starting from zero; server configuration from previous voting must not be used. The same should be done also after the public testing period.

**Freezing the functionality of VFS and VSS at the end of voting**

Vote receiving function of the Central System must be closed the moment the e-voting ends.
Writing into data files (logs) and data base tables that are open during e-voting must be blocked, the vote receiving applications of both VFS and VSS must stop working. Total separation of VFS/VSS from public network is recommended. A copy with limited data (only digital signatures of votes) and functions could be made of them for vote checking application.

**Destruction of votes after the final confirmation of election results**

We cannot be sure whether the encryption methods used now will still be functional after 30 years. Thus, after final confirmation of election results, all votes given must be deleted from all data carriers or these data mediums must be destroyed.
By that moment, the votes will be in:

– VSS (data base);
– VCA server;
– server audit copies;
– VSS – VCA transport CDs;
– auditing system.

Digital signatures, logs etc. may be preserved. Only the cryptograms of votes must be deleted.
**Technical supervisors of the system** must be constantly available during the e-voting period.
**Elections info line** should be ready for the function of technical support of voters and should be able to solve more frequent problems connected with ID card and VA.

## *4.3. Requirements for the organisation of voting*

### 4.3.1. Integration of the processes of e-voting and conventional voting

Conventional and e-voting processes must be integrated into common work process. It is also necessary to consider the election procedures where tendencies exist at present to break the rules, and reflect whether similar problems can arise in e-voting procedures.
Immediate forwarding of e-voting annulment lists at polling divisions must be motivated in some way.

### 4.3.2. Requirements for the description of procedures

All procedures necessary for e-voting must be previously described and tested according to these descriptions.
The documentation must include:

– conditions necessary for starting the process;
– end result to be achieved;
– process initiators and participants;
– technical activities carried out;
– necessary documents formed during the process and records on conducting the process;

– list of criteria for the success of the process.

NEC must appoint the persons who carry out and are responsible for the following activities:

– system development coordination;
– description of voting procedures;
– management and publication of system documentation;
– organising data exchange between e-voting system and NEC, *Andmevara* etc.;
– administration of Central System and co-ordination of administration;
– monitoring Central System during e-voting;
– keeping of system backup copies and auditing information;
– later archiving of digital signatures, auditing results and records of procedures;
– key management;
– guaranteeing technical support for voters;
– solving technical protests;
– conducting the pre-voting preliminary expertise;
– conducting interim audit during voting;
– conducting the audit of procedures following the voting;
– solving emergency situations;
– public relations.

The classical principle of division of roles, according to which different persons deal with system development, technical management, use and control, must be followed. It is necessary to appoint responsible persons, define communication channels and rules for escalating the problems in emergency situations.

#### 4.3.3. Requirements for the publication of system documentation

As much as possible of the system documentation must be public.

Concept diagrams and design decisions of the system, incl. this security analysis, should be public.

Voter application must be public and subject to authentication.

VCA public key must be public and subject to authentication.

Lists of candidates must be available in authentic way. This does not automatically mean their publication (on the web) – it is important that the interested person would have the possibility of accessing the full list.

Protocols used in public network (the protocol between VFS and VA) must be public. In principle everybody who wishes so must have the possibility to write his or her own personal voting application on the basis of public specifications.

The source code of all software components written for the system must be available for auditing; conditions for access are determined by NEC.

Informing of voters about the e-voting web page must be well organised. The address of e-voting web page must be published in public media and printed on the voter's polling card. The direct URL of e-voting web page, and not the general address of NEC web page, must be distributed.

In addition to that, the description of the checking of server certificate and the correct checksum must be published.

#### 4.3.4. Quality of service agreements

E-voting system manager must sign quality of service agreements:

– with the providers of Internet connection;

– with the providers of certification service (at present *AS Setrifitseerimiskeskus*) on receiving the annulment lists of certificates;

– with *AS Andmevara* on updating lists of voters;

– with the providers of validity confirmation service on operability of the service.

Besides that, there must be at least informal agreements with larger ISPs, because the access of their customers or voters to the e-voting web server depends also on them.

#### 4.3.5. Security control during system development

In addition to this preliminary analysis that evaluates the concept, the security of the actual realisations of system components must be analysed and tested.

#### 4.3.6. Pre-voting expert opinion on security

Before e-voting it must be evaluated whether the technical environment is sufficiently secure for e-voting. 21. The first years of the 21st century have vividly shown how rapidly the security of the Internet may decline. If security problems arise in connection with some specific technology used, e-voting should be cancelled or, if possible, the security loophole fixed (for example, firewall platform replaced with another).

#### 4.3.7. Interim audit during voting

*Interim audit* of e-voting must take place after first counting of votes but before confirmation of election results. Its purpose is to ascertain with the help of quick tests whether there have been gross security violations during voting, and if yes, then what kind of violations.

At least the following should be done during interim audit:

– Compare the lists sent to polling divisions, annulled votes and NEC annulment list.

– Compare restored votes and NEC restoration list.

– Check the logs of IDS and other security systems.

– Check the integrity of all Central System servers with the help of relevant application.

– Check whether each VFS log (Log1) item has a corresponding item in VSS logs (Log2, Log3).

– Check whether each VSS log (Log2, Log3) item has a corresponding item in VFS logs (Log1).

– Check whether each VCA log (Log4, Log5) item has a corresponding Log3 item.

– Check whether each VCA log (Log4, Log5) item has a digital signature in VSS.

– Check whether the sum of votes equals the number of rows of Log5.

We also recommend control counting of e-votes. It is similar to counting of votes, but it is conducted by other persons, (preferably) different security module is used and the results must be compared with the results of previous countings.
Interim audit must be done by technically competent persons who

– are not connected with ascertaining the final result of e-voting;
– are not connected with the development or management process of the system.

Written and signed act shall be prepared on the results of the audit.

#### 4.3.8. Post-election audit

The main function of the audit is to check whether all prescribed activities of voting process have been executed and recorded.
From the viewpoint of security, the post-election audit is a possibility to evaluate the security of system and, if necessary, make proposals on changing the security measures and security processes of the system for the next elections.

## *4.4. Key management*

The culmination of all elections is the counting of votes. It is done solemnly, by committees consisting of several members, under the watchful eye of observers and the whole society.
In e-voting, the high point is activating the vote counting application. Instead of solemn opening of ballot box, there is the VCA private key activation procedure where technicians with ponytails and committee members with bow ties side by side spell out numbers from a screen.
In the same way as the security of the ballot box starts from the workshop of the carpenter who made it, the security of e-voting starts from key management procedures that are done long before voters are invited to NEC web server. At first glance, key management procedures seem extremely complicated, their result is intangible, and even the smallest mistake creates error situation or makes them insecure. Therefore it is necessary to be careful and thorough in describing, following and auditing these procedures, and what is the most important – to understand every moment what is being done.

#### 4.4.1. General requirements

VCA key management has three absolute requirements that are the basis for all others. If these requirements are not met, e-voting will fail.
**Requirement for the authenticity of VCA public key:**
VA must contain correct VCA public key.
**Requirement for absolute operability of VCA private key:**
VCA private key must not under any circumstances be destroyed or become unusable.
**Requirement for absolute confidentiality of VCA private key:**
VCA private key should under no conditions become public.

### 4.4.2. Requirements for key management procedures

Several authorised persons must be present to carry out key management procedures. Hereinafter they shall be called *key managers*.

Key managers are personally appointed by the Chairman of the National Electoral Committee.

It must be guaranteed technically that VCA private key cannot be created, used, transported or destroyed (hereinafter we shall call these operations *using the key*) without the participation of key managers.

At the same time the absence of one (or several) key manager(s) must not hinder using the key, otherwise the risks connected with the persons of key mangers would be too great. Or:

Several key managers must be present for using the key, but it must not be a rule that all of them have to be present.

Traditionally key mangers must be independent, i.e. belong to different organisations.

A representative (or representatives) of NEC must certainly be among the key managers, but it should never be so that the key can be used with the help of NEC representatives only.

Besides key mangers, observers must also be present when the key is used.

A written record signed by all key mangers present must be prepared on each operation of using the key. It must contain at least the following information:

– participants;
– time (period) and venue;
– what was planned to do;
– what was actually done;
– final result of the activity;
– problems that arose.

Acts are archived by NEC.

In test and development systems keys that differ from the ones of the actual e-voting system must be used.

VCA key pair must be RSA key pair with the length of 2048 bytes. At present, shorter key is not safe any more, but longer key makes operations too slow.

VCA private key and its components must never exist in open (unencrypted) form.

A backup copy (or two) must be made of VCA private key.

The same requirements that apply for private key also apply for backup copy(ies).

When the voting results have been confirmed, VCA private key and its copies must be destroyed.

VCA public key must be distributed in the form of self-signed certificate.

The certificate and the information necessary for controlling it must be public.

Essentially the above requirements mean that hardware security module (HSM) must be used for VCA key management. In that case VCA private key is kept only in the static memory of the security module. For using the private key, the security module must be authorised with the help of several special chip cards and PIN code. Key managers are the owners of these chip cards and know the PIN codes.

### 4.4.3. Patterns for appointing key managers

There are several possible patterns for controlling the access to VCA private key. They all realise the requirement "several key managers must be present for using the private key, but all key managers do not have to be present".
Generally we presume here and hereinafter that all key managers possess chip cards and corresponding PIN codes with which they technically realise their right to use the key.

**Many-of-many (M-of-N) pattern**

There are altogether N key managers.
For using the key, the presence of M persons of them is necessary, whereas M<N.
For example: if N=5 and M=3, there are five key managers and for using the key, the agreement between and the presence of three of them (naturally with chip cards and PIN codes) is necessary.
The advantages of this pattern are logic, effectiveness (the number of persons and cards needed is small) and relative reliability: in the cast of 3/5 any two key managers may be absent and the key can still be used.

**Pattern of sets of keys**

A set of N different chip cards is necessary for using VCA private key.
There are K copies of each different chip card, each one of them held by different key manager.
Thus there are altogether N×K chip cards and key mangers in the system.
It can also be said that there are N main key managers, each of whom has (K–1) deputies.
For example: `N=3`, `K=3`, the total number of key managers is `3×3=9`.
The scheme is similar to opening a door with several locks: each lock of the door has a different key, there are several copies of each key; one key for each lock is needed to open the door.

### 4.4.4. Key management procedures

We cannot give the descriptions of key management procedures in this analysis because they depend on the model of security module used, and presenting detailed procedures would essentially be choosing the brand of security model. Therefore the procedures presented are general, offering one of the possibilities for meeting the requirements defined above.

**Creating VCA key pair**

VCA key pair must be created before the beginning of e-voting. One part of it, VCA public key, must be integrated into VA, and this takes time.

– Key managers authorise security module, using chip cards and PIN codes.
– Key managers give security module an order to create VCA key pair.
– Security module generates private key and public key.
– Security module saves private key in its static memory.
– Security module generates VCA public key certificate.
– Security module prints out the certificate or public key.

The printer used is connected to the security module directly, without the mediation of

a computer.
The printout is signed. This is the so-called “original” of the VCA public key.

– Security module shall save public key into a file. An alternative is copying the key by hand from the security module console.
– Security module is brought into ordinary, unauthorised regime.
– The certificate contained in the file is printed out and compared with the original.
– Checksum is calculated for the certificate.
– Certificate and its checksum are made public.

**Creating backup copy for VCA key pair**

This procedure will create a copy of VCA private key in the memory of another security module.
The security module may export private key only when its parts (components) are written on separate chip cards. These chip cards are meant only for transporting this key and they shall be destroyed after the procedure.

– Key managers authorise the security module.
– Key managers give security module an order to export VCA private key.
– The security module exports the key to chip cards by components.
– Security module is brought into ordinary, unauthorised regime.
– Chip cards are taken to another security module situated in the same room.
– Key managers authorise the other security module.
– Key managers give security module an order to import VCA key pair.
– The security module shall read the components of VCA private key from chip cards, asking the PIN code of each chip card.
– The security model shall calculate the public key corresponding to the private key that has been read.
– Security module is brought into ordinary, unauthorised regime.
– Security module prints out the calculated public key.
The printer used is connected to the security module directly, without the mediation of a computer.
– The printout is compared with the original of VCA public key.
– Chip cards are physically destroyed.

When the printout and the original correspond to one another, both security modules have the same VCA key pair. In addition there is a digitally signed file with VCA public key.
After these procedures security modules can be disconnected from the power network and put into the safe; next time they are needed is when the results are counted.

**Testing VCA key pair**

Encrypting the vote is the most non-transparent part of e-voting. All other operations – displaying the web page, signing the vote, activities going on in VSS – can be controlled in different ways, but the correctness of encryption of the vote is hidden until the moment the votes are counted.

Therefore it is necessary to check separately after the final completion of VA whether the whole process functions and if VA contains the right public key. For that, it is necessary to give one or more votes with the help of VA, and check if they can be opened by VCA and the result is correct.

– One or several correct test votes are formed with the help of VCA.
– Besides that, several faulty votes are formed, some of which contain forbidden data and some are not encrypted with the correct key.
– Test votes are copied into VCA server.
– Security module is connected to VCA server.
– VCA application is started.
– Key managers authorise security module, using chip cards and PIN codes.
– VCA application opens the votes (calculates the result of voting).
– Security module is brought into ordinary, unauthorised regime and disconnected from the VCA server.
– VCA server is booted.
– Counted result of VCA is compared with the votes formed.

**Counting the votes – using VCA private key**

By that time there must be a file in VCA server that contains encrypted but no longer signed e-votes (the so-called "inner envelopes"). The authenticity and integrity of the file must be guaranteed – i.e. transport of the file from VSS to VCA server must have been procedurally correct and the file must have been compared with VSS output, for example with the help of checksum.

– Security module is connected to VCA server.
– VCA application is started, its input is the e-votes output file of VSS.
– Key managers authorise the security module.
– VCA application opens the votes and calculates the result of voting.
– Security module is brought into ordinary, unauthorised regime and disconnected from the VCA server.
– VCA server is booted.
– File of votes is deleted from VCA server.
– The contents of results file, Log4 and Log5 are checked.

As we can see, the counting of results is relatively simple procedure in comparison to creating the keys. And so it is: most of the complications with key management are connected with creating and distribution of keys, not using them.

**Destruction of VCA private key**

When the voting results have been confirmed, VCA private key must be destroyed. Relevant procedures are given in security module instruction. The producer of security module is responsible that the keys situated in the security model are irretrievably destroyed.

**Integration of VCA public key (certificate) into VA**

VCA public key is a part of VA. Risks caused by wrong public key in VA are described under the risks of key management.

After the final completion of VA but before putting it into Web server, it must be checked separately whether VA contains correct VCA public key. The procedure "Testing VCA key pair" that is described above can be used for that.

## *4.5. Summary of measures*

Let us sum up in briefly all presented security measures.

### 4.5.1. Technical security measures

For guaranteeing general security, we recommend to use the measures corresponding to security class H of three-level IT baseline protection system.
To guarantee transparency, the design and documentation of the system must be as public as possible.
*Central System* must be a separate system with zoned network, firewall and intrusion detection that does not vitally depend on any outer data source except the voters themselves; when the changes to the concept offered here are realised, neither updating voters' data base nor getting time stamps are no longer time-critical. In all components the purpose of design is simplicity, restricting of properties and separation of functions. The objective is checkability; everything that does not endanger the privacy of votes given must be logged and saved; logging must be duplicated in separate servers; monitoring faults and intrusions at both network and application level is strictly recommended.
More stress must be laid on the correctness of software and management than on functionality and capacity. Fighting against denial-of-service attacks is not included in the functionality of the system, prevention of such attacks is guaranteed by means that are not part of the system.
Due to the small space of data and system, restricting the scope of data losses and quick restorability of the system can be achieved with relatively simple measures. Constant availability of system managers is also an important requirement.
*Data* formats must be kept simple, but there should always be a possibility to check the correctness of data existing or moving in the system. In the case of data sources and receivers from outside the system the integrity check must be especially thorough. Some data should be published in controllable way (see Annex 2, "Data channels to the system and from the system"). Cryptograms of votes must be destroyed after confirmation of election results.
The qualities of *voter application* arise from the conventional requirements for signing application. The most complicated aspect is the need for independent verification report – for that the application must either get from the server or possess itself the permitted data of candidates. Besides that, displaying the data of candidates must take place independently of the web server, so that the server would not know whose data was examined.
Special attention must be turned to the security of *vote forwarding server / web server* as a public and thus the must vulnerable server. Key words are again simplicity, restricting of properties and conservative programming. A special requirement is the need to connect the person of SSL user and the need to limit the information on

candidates. VFS must give user application feedback on taking into account of the vote sent.
*Vote saving server* as the most complicated component is essentially divided into two: data acquisition motor functioning during vote giving and subsequent sorter/canceller. Those functions should be separated as much as possible. Moreover, the database which should be guarded with conventional measures (access restrictions, adequate rights, logging, etc.) is situated here.
VSS data base application faults may enable access to data and ignoring restrictions, therefore the fault-freeness of VSS applications is also a security requirement.
*Vote counting application* where the votes are in open form and which actually calculates the result of voting is protected mainly with physical security and key management. Besides several VCA technical protection measures we recommend to print the output of VCA – voting result – directly from VCA.
We listed the data collected into the *auditing system* and also wrote down some necessary control activities that should be conducted. We also stress that the data of auditing system must be protected the same way as the data in working servers.
At first glance *key management* is an unusually complicated subject that connects organisational security and cryptography and therefore more than average attention should be paid to describing it. The analysis suggests a key management scheme based on safety modules, lists procedures connected with key management and the activities carried out in their course.

### 4.5.2. Organisational security measures

The most important security measure is appointing the *executive and responsible parties* of e-voting processes. Here the *principle of division of roles* should be observed and the responsibility divided among participants in accordance with the roles of conventional voting and the national practice of information systems management.
Rules must be established for *solving special situations*, *informing* and *escalation of problems*.
The workflow of e-voting and conventional voting should be analysed and it must be guaranteed that they complement and not disturb one another. The procedures must be described and tested, at the same time the danger of overformalising the processes must be avoided.
E-voting must be accompanied by *informing voters of the safety of e-voting*, emphasising the authenticity control of web page and safeguarding the security of the voter's computer.
*Constant security control* over the development and implementing of e-voting system must be guaranteed.
Before each e-voting, *pre-voting expert opinion of security* must be obtained.
Before confirmation of election results *interim audit* must be carried out during elections.
We also recommend *control counting* of e-votes.
After confirmation of election results, *post-elections process audit* must be carried out.

*Quality of service agreements* must be signed with important outside parties.
*Voter technical support information line* must be available during the voting period.

## *4.6. Risks that are to be accepted*

### 4.6.1. Need to spend resources on organisational and technical security

Security is an expense. It is in conflict with effectiveness, convenience and simplicity; operations must be duplicated, monitoring and follow-up control added to activities, security analysis added to development and auditing to management. All this takes labour and money.
It must be accepted in advance that these resources should be found.

### 4.6.2. Possible insecurity of the voters' computers

We think that the possible insecurity of a certain number of the voters' computers is an acceptable security risk from the standpoint of e-voting. The main argument here is that the parties who have the knowledge, resources and access necessary for attacking the computers of a large number of voters do not have a motivation for that; and the political forces who have the motivation cannot take the risks connected with such an intrusion.
People who conduct business and financial affairs through computer are in "greater danger" every day than during e-voting.
There are no reliable methods for alleviating risks connected with AIPs, and such risks simply have to be accepted.

### 4.6.3. Need to trust Central System computers

It is necessary to accept the fact that the components of the systematic layer of Central System computers simply have to be trusted. Obtaining these components from trustworthy sources diminishes the risk.

### 4.6.4. Impossibility to support all voters

### 4.6.5. The fact that only the users of more widespread personal computers can e-vote must be accepted. At the moment it is possible to develop HR for Windows, Linux and MacOSX platforms, but not all possible versions and operating systems. Besides that, the list of platforms supported by ID card basic software sets its limits. Concentration of risks and the possibility of negative media report

Concentration of risks must be accepted. Instead of frequent small errors of human procedures, rare but large-scale faults with high media value arise in the technical system.

### 4.6.6. Risks arising from formalisation of processes

The rules of physical world are always "soft" because the relations between human beings can always be adapted to each special case. In information systems, however, the rules are rigid and it is not possible to ignore them or "cut corners".

As a result of that, overformalised processes may start to hinder work so that it is not done at all any more, or some essentially feasible ways of making exceptions that have worked so far may be lost.

# 5. GENERAL EVALUATION OF THE CONCEPT

**Correspondence to the election requirements**

In the "Requirements" chapter we described the contradictions between the requirements for elections and said that it is necessary to find a compromise where all main requirements are met and the risks taken are accepted at political level. Taking into account the existing risks and the global information security situation, we find that the solution for e-voting continues to be secure and reasonably grounds these risks.

Here we bring the solution corresponding to the requirements presented in chapter 2.2 "Requirements to be met"

| Requirement | Way of guaranteeing |
|---|---|
| Authorisation and authentication of voters | Guaranteed by system design. ID card or Mobile-ID is stronger way of authentication than showing a paper document. |
| "One person – one vote" | Guaranteed by system design. |
| Prohibition of falsification of votes | Guaranteed by system design and auditing. Digital signature is unfalsifiable, other errors are identified during interim audit. |
| Uniformity of voting | The evaluation whether the possibility to vote with the help of computer violates uniformity or improves it does not belong to the scope of our analysis. |
| Possibility for electronic re-vote | Guaranteed by system design. |
| Supremacy of conventional voting | Guaranteed by law and the general organisation of the elections. |
| Annullability of vote by the voter | It is not a requirement but is indirectly guaranteed by the possibility to change one's vote |
| Possibility to give an empty vote | Is not a requirement, is not guaranteed. |
| Secrecy of vote | Guaranteed by strong encrypting and key management. |
| Privacy of the fact of voting | Guaranteed softly – monitoring network connections by their providers is theoretically possible. |
| Unprovability of voting | Guaranteed by the properties of VFS/VSS applications and the possibility to change one's vote by conventional voting. |
| Operability of the voting system | The scheme is modular and as simple as possible. Technical operability is no problem, software failures should be avoided by testing. |

| | |
|---|---|
| Transparency | Guaranteed by the simplicity of design, publicity of system principles and controllability of source code of applications |
| Auditability | Guaranteed by system design. Technical realisation has been created through logging, audit system and audit application. |
| Controllability of calculation of votes | Guaranteed by voter feedback possibility and audit application. |
| Repeatability of counting | Guaranteed by system design. |

**System architecture and simplicity of solution**

The first security requirement is the simplicity of the analysed system or process. Complicated dispersed information system with many applications and connections between them always contains more faults than simple and comprehensible system. We find that system architecture is reasonable and suitably modular. In the course of analysis several possibilities were offered for changing that but during the discussion it was decided to discard them or use them only as abstractions for making the analysis easier.

We stress that the analysis was made on the assumption that e-voting info system is separated from other election info system and all e-voting info system communication with the outside world will take place through very limited interfaces. When uncontrollable information channels between e-voting system and the rest of the world emerge (for example, common servers with conventional elections applications are used to save expenses), the security dangers connected with that are also uncontrollable.

**Realizability**

The technical solution of e-voting has been realised in Estonia on the basis of local IT knowhow, and it has been used in practice altogether five times. Thus by today e-voting has become an everyday reality and is not just a concept any more.

**Compatibility with the European Union recommendations**

The concept is in harmony with the future e-voting requirements of the European Union or IP1-S-EE working group recommendation document [IP1-S-EE].

# 6. CONCLUSION

Solving the controversial problem of electronic voting was an interesting challenge in 2003. Now it can be said that that the challenge has been solved and realised in practice.

The existing solution is quite simple. The scheme that is mathematically safer but more complicated from the standpoint of realisation makes the solution more complicated, increases the number of components and the connections between them and, as a final result, reduces security. In the choice between the theoretical security of voting scheme and the complexity of its realisation there has to be an optimum, and the technical solution corresponding to it gives the best compromise between the requirements set.

The strong points of the Estonian voting pattern are:

– comprehensibility and similarity to conventional voting;
– maximal use of the digital signature solutions available in Estonia (ID card, DigiID, Mobile-ID);
– using only simple encrypting algorithms,

and last, but not least –

– it can be realised with the help of IT knowledge existing in Estonia.

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.

We believe that the risks of the described voting pattern are managed so that the possibility of the dangers becoming a reality or the damage caused is acceptably small. It can be said that by putting different parts of the system to distrust and monitor each other and adding control by humans where necessary, we achieve sufficiently secure e-voting system.

Naturally organisational measures have to be used besides the technical measures (cryptography, intrusion detection, double control of data, etc.). division of tasks and responsibility, formal procedures, awareness and managing of risks by NEC, prepared action plans for solving emergency situations, independent audit.

We believe that the security of the new e-voting mechanism is higher than that of the conventional voting using ballot papers. In the future, too, it requires well-planned technical solution, careful development work and – what is the most important – responsible use, but all systems that are as critical require that.

# 7. ANNEX 1 – DATA PROCESSED IN THE SYSTEM

We will recount the data processed during e-voting.
Naturally, the system also includes large amounts of secondary information which is not touched on here: user passwords, source code of applications and system documentation, microchip cards of key managers, etc.
*Location* in the following table means that the information is accessible to a particular component of the system at a certain time. *A* is the computer of the voter; *audit* – auditing system; *paper* – information issued or available on paper. Internet is not a component because the data moves in encrypted form and cannot be read by the communication channel.

| **Information** | **A** | **VFS** | **VSS** | **VCA** | **audit** | **paper** |
|---|---|---|---|---|---|---|
| **Input information** | | | | | | |
| Polling list | | x | x | | | |
| VA | x | x | | | | |
| Personal information on voter/ digital signer | x | x | x | | x | |
| PIN-code of voter's ID card | x | | | | | |
| **Information generated in the voting process** | | | | | | |
| VCA secret key | | | | x[1] | | |
| VCA public key | x | x | | x | | x |
| Signed, encrypted votes | x | x | x | | | |
| Encrypted, unsigned votes | x | x | x | x | | |
| Digital signatures | x | x | x | | x | |
| Unencrypted votes | x | | | x | | |
| Confirmations of acceptance of votes | x | x | x | | | |
| Lists of e-voters to polling divisions | | | x | | x | x |
| Applications to nullify by committees | | | x | | x | x |
| NEC applications to reinstate nullifications | | | x | | x | x |
| Voting results | | | | x | x | x |
| **Logs** | | | | | | |
| LogWeb – access-log of web server | | x | | | x | |
| Technical error logs of applications | | x | x | x | x | |
| Log1 – votes cast | | x | | | x | |
| Log2 – votes nullified at sorting | | | x | | x | |
| Log3 – votes accepted, sent to counting | | | x | | x | |
| Log4 – faulty votes found at counting | | | | x | x | |
| Log5 – votes counted in voting results | | | | x | x | |

1) VCA secret key actually located in the security module, and is only open to applications for using, it cannot be copied through VCA server.

# 8. ANNEX 2 – DATA CHANNELS INTO AND OUT OF THE SYSTEM

**Input information**

| | | |
|---|---|---|
| list of candidates | NEC | → VSS |
| list of voters | Andmevara | → VSS |
| renewals of list of voters | Andmevara | → VSS |
| nullifications and reinstatements | | NEC → VSS |
| vote | voter | → VA |

**Output information**

| | | |
|---|---|---|
| feedback | VA | → voter |
| voting status of a voter | VA | → voter |
| lists of e-voters | VSS | → polling divisions |
| results of e-voting | VCA | → NEC |
| monitoring results | supervisory application | → NEC, system supervisors |
| logs, audit results | auditing system | → NEC, archive |
| archived digital signatures | VSS | → archive |
| e-votes | VSS, VCA | → to be destroyed |

**Information for publication**

The following information must be public, with guaranteed authenticity and comprehensiveness.

- candidate list and additional information
- URL (address) of e-voting web server
- VA, its signature or checksum and control method
- public key of web server, its checksum and control method
- public key of VCA, its checksum and control method

All who so wish must have access to the following, on conditions determined by NEC:

- VA–VFS communication protocol
- technical documentation of the system
- source code of system components

# 9. ANNEX 3 – TECHNICAL RISK ANALYSIS

We can formally say that the risks of comprehensiveness and confidentiality are defined by the comprehensiveness and confidentiality of all information over all components in the system. Risks of operability are also easily identified by the efficiency/ operation speed of each component, application or data carrier. The objective of the following risk analysis is not to list all possible versions – a simple cross table of system data and components would suffice for that, while the outcome would be of insignificant practical value – but drawing attention to the most important points.

## *9.1. Classification of risks*

There are many possibilities for classifying and presenting the risks: according to the attacker (roll-centred position); by the affected part of the system (architectural position); by the cause (error, attack, impact of environment); by the data in risk of exposure (data-centred position); by the chronological order (process-centred position); by critical status; etc.
Our risk classification is based on the requirements set by the system.
In other words, we divide the risks into classes according to the security attribute under attack.

– *Risks to the integrity* jeopardise the veracity of voting results;
– *Risks to the confidentiality* jeopardise the confidentiality of voting.
– *Risks to the operability* jeopardise the operability and usability of the system.
– *Risks to the reliability* jeopardise the correctness of the e-voting process.
– In addition we point out *the risks of key management*, which compose a logical entity.

All risk assessments are qualitative.

## *9.2. Risks to the integrity*

### 9.2.1. Discrimination errors

*Discrimination errors* are errors whereby the e-voting system treats some voters differently form others. It could also be called selective operability but since it jeopardises the voting uniformity requirement, it jeopardises correctness (integrity). This type of error may occur in any component of the system. Discrimination may be random (then it is just a quality-related error) or introduced into the system on purpose.
Possible examples:

– Web server does not allow contact from a certain county, such as Virumaa.
– The central system VFS incorrectly logs the hashes of certain votes and the later check annuls such votes.
– The central system is unable to check digital signatures of signatories whose name includes the letter *õ*, and rejects such votes.

– The voter application does not function in the Russian language Windows, leaving non-Estonians unable to vote.
– The voter application does not work in the Mac-type computers of the Academy of Arts, leaving art students unable to vote.

The discriminatory risk in itself is not great, as the elections are organised in a way which allows all voters to vote in the conventional way. At the same time, discrimination may lead to various reliability risks.

### 9.2.2. Risks related to Internet usage

The normal data exchange between the Central System and voter browser or VA cannot be attacked via the Internet. The voter could, however, end up on (or be directed to) a web page imitating the e-voting page and deceiving the voter, or a page attacking the voter's computer and thus gaining control over the e-voting process.

*Directing the voter to a falsified web page*

The voter could end up on a falsified page:
– through false information,
– input error (e.g. typing an address like ww.wvvk.ee),
– technical reasons (DNS errors/attacks, wrong configuration of the voter computer, ...)

The recent successful attacks against Internet banks have been committed by way of forged web pages and massive wrongful notification of bank customers. One such attack also occurred in Estonia (and failed only thanks to the insufficient language skills of the hackers).

*Man-in-the-middle attacks between the web server and the voter application*

Man-in-the-middle attacks constitute another special case whereby a fake web page intermediates the whole communication between the VA and the web server. Communication channel man-in-the-middle attacks are practically identical to web server attacks: someone creates a fake web server for a certain part of network, which enables them to feed to the voter a fake application, detect his or her choices, etc. There is generally no good solution against man-in-the-middle attacks. The classical method – authenticating the web server through a server certificate – requires an informed and careful user. Fortunately the Republic of Estonia has the advantage of having chip cards, thanks to which we can request that the client identifies himself or herself in the web server, which completely prevents man-in-the-middle attacks. In the case of Mobile-ID, the mutual authentication of the SSL channel cannot be carried out at equal level, but the voters using Mobile-ID do not form a majority at the moment.

*Attacking the computer of the user and gaining control over it* is another risk inextricably linked to the use of the Internet. The situation is all the more precarious due to the security failures in the Microsoft software (in operating systems as well as the browser) published during the recent months. Since the voter's ID-card is also accessible in the computer at the moment of voting, the e-voting time might be the most attractive for attack.

Internet can also violate the equality of the voting process through *discrimination* or selective obstruction of voters. For example, it may happen that on the very e-voting days network connection is down in Ida-Virumaa.

### 9.2.3. Web server/VFS

Contents of the web server – lists, voter application, static information – are correct at the start of the e-voting. These data are similar to the lists, ballot papers and guidelines at the polling division and must be checked before the start of the elections. This content can be changed without authorisation – either by breaking into the server or by the technical administrator of the server. Since this distorts the most important input data of the voting process, the result of the voting process, i.e. the vote, will also inexorably change.

*Jeopardising voter application*

The voting application on the web downloadable by voters. The result of its unauthorised changing is large-scale loss of e-vote security. Votes may be forged, vote and voter privacy violated, some candidates shut out from voting, etc.

*Unauthorised change of candidate list*

Web server houses a list of candidates displayed to voters. If it is not correct, it cannot result in a correct vote. As a result of unauthorised change of data (exclusion, inclusion or interchanging of candidates, changing information on divisions, etc.) the votes of all persons who have received the changed list will become incorrect. The voter could not choose the desired candidate, signed an unintended choice, etc.

The change of the web server programme to the effect that a wrong list is forwarded to the voter upon request has the same result.

*Static content of the web server jeopardised*

The web server houses the static notification data necessary for e-voting ("To access e-voting click the red button"). Unauthorised changing of this information (e.g. by adding campaign materials of a political party) will cause problems but most probably will not jeopardise the integrity of the voting result.

*Errors in equal display of data on candidates*

The system should display the data of all candidates or lists in a similar manner and ensure that the visual side of the application does not influence the process of making a choice. The most serious problem is probably the candidates who remain "off-screen". The Estonian alphabet might also cause display problems in the computers of certain voters. Again this risk is magnified by the fact that the environment showing images to the voter, his or her computer and browser, are unpredictable.

By displaying data on a screen with variable size, it is extremely easy to create the so-called Florida Butterfly Ballot, where the names of candidates and the boxes to be marked do not line up, and confused voters may vote for the candidate they did not want.

*Classical web applications and web server errors*

These are last on the list because they are well-known and typical, not because they are safe. *Cross-site scripting, session fixation attacks*, mistakes in checking input data, *code/SQL injection*, provision of configuration data through error messages, etc. – web applications suffer from a large number of errors committed again and again. A

relevant list could be found at [OWASP] page, among others. Their frequency is caused by the simple fact that they can only be avoided through nauseatingly careful programming which is unfortunately a rare luxury in our rushing e-world.
Web server itself can also be managed incorrectly in a hundred ways which expose its contents to attacks. The attacker could thus gain control over the secret key of the HTTPS certificate of the web server and carry out man-in-the-middle attacks that are much harder to detect.

#### *9.2.4.* Voter's computer, web browser, VA

*Exposure of candidate list*
If the candidate list displayed to the voter is not correct, neither can his or her vote be correct. If the numbers of two politicians are interchanged or one of the candidates is removed completely, voting results are clearly incorrect.
*Exposure of voter application*
Exposure of the application changes the voting process, with unpredictable results (falsification of a vote, loss of its confidentiality, impossibility to choose certain candidates, etc.)
What is the difference between attacking the list (data) and the application? It is generally much simpler to attack the data. Data displayed on the web browser as an HTML page can easily be changed by taking advantage of the browser security loopholes, without having to gain full control over the voter's computer. It is many times harder to modify the application (or information displayed by intra-application means).
*Substitution of the VCA public key in voter application*
This is in fact a special case of jeopardised VA, described under risks of the key management.
*Functional failures of the voter application*
An application may contain design errors, inadvertent errors as well as intended troy horse type of features. Thus VA could:
– substitute the voter's choice with something else;
– not display the names of certain candidates;
– simply "refuse to work" in some conditions.
It is in fact certain that the computer/OS/browser combinations "supported" for VA do not cover all voters. Functional errors are thus almost 100% probable and the question remaining is whether they occur sufficiently rarely and whether they violate only the integrity of the elections (do not support Russian-language Windows versions) or the integrity of the results directly (do not display the candidates of certain parties, change votes).
*Use of misleading data by the voter*
The fact that the voter votes without leaving his or her normal environment creates the risk of misleading publicity campaign. In this case the voter receives (by e-mail, snail mail, etc.) misleading publicity materials. He or she could thus receive a message "Vote for P.P., number 666!" although the number 666 on the candidate list is T.T.

At conventional elections, the voter receives correctly defined data in the polling division and therefore does not depend on publicity materials in such a way. E-voting increases this risk.

*Intentional sending-off of an incorrect vote by the voter*

VA functions in the computer of the voter and is thus under his or her control. This means that the voter can – if he or she has sufficient technical knowledge – change its behaviour according to personal preferences. It is in principle possible for a voter (or someone else) to write an alternative application to substitute an official one. It is not bad in itself, as is nothing deplorable in the fact that people use different web browsers – it is only important that the same standards are supported (HTTP, HTML, CSS, etc.).

This means that we should make no assumptions about the correctness of the vote sent to VFS by the voter. The vote might be unencrypted or unsigned; encrypted with the wrong key or signed with an unofficial certificate; include incorrect data (e.g. the name of the candidate instead of his or her number, or a political manifesto); be formatted incorrectly or be incorrect in any other way.

Continuing the web browser analogy – HTTP request is also completely under the control of its sender and no assumptions should be made about its contents and format. Wrong HTTP headers, suspicions about the data in HTML forms, buffer surcharges and other attacks are the facts that have to be taken into account with every web application. The e-voting server side application must also mistrust the formulation of the sent vote

*Signing of the vote with an invalid (annulled or suspended) certificate*

It is possible for the voter to sign the vote with an annulled certificate (e.g. with a stolen card) or annuls the certificate after the vote is sent.

*General risks of signature application*

The voter application has all the classical risks of signature application, deriving from the fact that it has access to the voter's ID-card. The application might thus sign something else besides the vote, or e-mail the PIN-code of the voter's ID-card to a hacker.

While in other cases such errors could be detected through server feedback ("You signed a loan contract for 1 million dollars. Thank you!"), the vote confidentiality requirement excludes this possibility – VFS may not tell the voter "You voted for candidate No. 666".

### 9.2.5. Intranet

The network (or firewall) can violate the integrity of the voting results through *discriminatory* forwarding of votes.

*Changing the list of unsigned votes*

The worst risk to integrity is changing the list of unsigned votes transported from VSS to VCA. It is possible to add an unlimited quantity of votes and erase authentic votes.

### 9.2.6. VSS

VSS as the most complex component in the functional sense has the best control over the votes, and thus also the best opportunities for manipulating with them. VSS could erase votes and add them, annul without reason, modify, etc.

All e-voting input data meet in VSS: polling and candidate lists, cast votes and their status (valid, faulty, immediately annulled), notifications of annulment and restoration. In the end these create a file of "anonymous" votes to be forwarded to the VCA.

*Errors of input data in VSS*

It is clear that errors in any data source directly impact the integrity of the result. If a candidate was missing even for a day from the candidate list, the voters were not able to vote for this candidate on that particular day. If someone is missing from or added to the polling list, then either a legal voter cannot vote or someone who is not a voter can.

*Functional errors of VSS applications*

Errors in VSS applications (receipt of a vote and checks carried out, annulment and restoration of votes, sorting) might influence the integrity of voting results in innumerable ways. If the fallacy of data is likely to remain unimportant (it is not probable that half of the polling list is missing), the scope of errors in the application and the consequences are unlimited.

A solution is multiple checking of VSS activities. This requires VFS logs, auditing application, etc.

*Digital signatures checking errors*

Digital signature checking algorithm used by VSS must be absolutely free of errors, to avoid large scale falsification of votes (through acceptance of false signatures) or violation of the integrity of elections (rejection of correct signatures).

VSS might thus accept votes signed with any certificate whose form (issuing body and distinguished name of the holder) resembles the official signature certificate. This is one type of functional errors of the VSS application.

*VSS exposure*

Exposure of VSS through either an attack or malicious activities of its administrator(s) might change the voting results similarly to the errors of VSS applications.

At the same time, technical security of VSS is better than that of the web server, since it is not accessible from the public network.

### 9.2.7. VCA

Vote counting application is the component of the e-voting system which counts votes and announces the actual result. It is therefore inevitable that every *functional error of the VCA application* is a direct error of the integrity of e-voting results.

### 9.2.8. Validity confirmation service

Validity confirmation service cannot influence the correctness of the voting in any other way besides *discrimination*. Even than it cannot refuse from giving validity confirmation, for example on the basis of the content of the vote, because the service

provider sees only the hash of the digital signature with no possibility of deriving any information on the vote or the voter.

#### 9.2.9. Auditing system and auditing application

This subsystem with a checking function cannot influence the integrity of the system.

## *9.3. Privacy risks*

#### 9.3.1. Web server/VFS

*Violation of the confidentiality of the fact of voting*
Access log in the web server contains loading times of the application, IPs and browser versions. If ID-card is used for authentication, the web server automatically learns the personal identification code and name of the voter. If division code request is logged on the basis of a personal identification code, the connection between the personal identity code and the division will remain in the server.. Similar information is contained also in VFS log.
*Timing of attacks against the user's computer*
Surveillance of the web server or user network connection allows the voter computer to be attacked in real time, at the moment of voting.
*Violation of vote confidentiality*
It is possible for the web server to detect the choice of the voter. This can happen, for example, when the web application design is faulty and some additional information (such as a photo) is asked from the web server in the VA when choosing a candidate.

#### *9.3.2.* Voter's computer, web browser, VA

*Violation of vote confidentiality*
*Violation of the confidentiality of the fact of voting*
The voter's computer is the first and the most likely source of leakage of the voter's choice and other data on the voter. In addition to the security problems of the voter's computer described above, a voting trace (trace of connection to the web server) will also remain in the user web browser log.

#### 9.3.3. Connection channel (Internet) between the VA and the Central System

*Violation of the confidentiality of the fact of voting*
Traffic monitoring permits to detect the computer from which the Central System is approached.
This is possible even if the connection between the VA and the Central System is encrypted. It is difficult to hide the fact of submitting a request to the web server and receiving a response in the approximate volume of the VA. If this is followed by the communication "vote – VFS confirmation", the fact of voting is fairly obvious.

#### 9.3.4. VSS, Intranet

*Practically all data in the e-voting system* – with the exception of the actual votes which are accessible for the VSS only in encrypted form – *can leak* from VSS and the Central System.

Also the *leaking of the complete data base of votes* is the most likely from VSS, which contains it its entirety (other components mediate the information during their working life). The danger lies in the fact that the technique used to encrypt a vote may not function in 30 years' time and the owner of the data base could then violate the confidentiality of all votes.

#### 9.3.5. VCA

VCA contains the voting results from a certain moment.
Since VCA knows the value of every encrypted vote as well as its hash, it is possible to use it to *link the hash of a vote and the value of the vote cast*. The hash of the vote would enable to find the caster of a vote from VSS.
For this, the attacker should monitor the memory used by VCA or take advantage of the errors in the VCA itself.

#### 9.3.6. Validity confirmation or time stamping service

Validity confirmation service checks the validity of the certificate of the voter, which means that a *list of persons who have voted electronically* and their time of voting is generated inside it.
*Use intensity information concerning the persons who have voted electronically may leak* through the validity confirmation or time stamping service server.

#### 9.3.7. System output

*Limited confidentiality of e-voting results*
Every division can use this formula for every candidate: votes cast at e-voting = final election result – votes cast by standard procedure.
The e-voting result is thus not a secret for the persons who have access to the election protocols compiled by divisions of the votes cast by standard procedure. This is a problem if the number of voters is small, but the problem of corruptedness of a committee member cannot be solved by technical means.

#### 9.3.8. Auditing system and auditing application

Logs are the classic place from which data is leaked. Auditing system contains logs that contain information on the votes cast and large amounts of technical information about the functioning of the system. This information must be protected as tightly as the data contained in VSS and VFS.

### *9.4. Operability risks*

Operability problems of standard systems can be broken down in proportions of 4:2:1 into management errors, software errors and technical failures of hardware. In the e-voting system management and software errors probably carry even more weight because of the relatively short operating time of the system.
Thus the main causes of operability problems are *management errors* and *untested software*, followed by *hardware failures* and *wrongful planning of necessary system resources*.

Operability infringements are the easiest to detect as requirements to operability can be defined quantitatively and the extent to which the system corresponds to them can be measured.

### 9.4.1. Voter's computer, web browser, voter application

Voter, and not the managers of the Central System, controls his or her computer and its software. The present analysis therefore does not view these as risks of the e-voting system.

*Operability of the voter application* probably turns out to be the Achilles' heel of the whole e-voting system. Quality problems are almost certain to occur – it is not easy to picture an application which functions almost uniformly in all clients' computers used on the Internet.

### 9.4.2. Connection channel (Internet) between the VA and the Central System

Generally the problems of the operability of the communication channel are to be born by the voter, just like the voter is responsible for his or her own arrival to the polling station.

There might be a problem with the *great volume of the VA* which stops it from being downloaded by voters with a slow Internet connection or makes the whole voting process too slow.

### 9.4.3. Web server / VFS, firewall, Intranet, VSS

These components participate in the voting process directly and thus hold the most critical significance from the viewpoint of the system reliability. Their failure causes e-voting to fail.

*Servers' and communications network's operability risks* are not specific to e-voting. These come up in the work of every organisation requiring a reliable information system and there are traditional neutralising methods for dealing with them (duplication of hardware and network connections, data mirroring, monitoring, etc.).

*Software failures* – A definite source of risk is the software developed for e-voting, the failures, error tolerance or random programming mistakes of which might not be thoroughly tested.

*Data base errors* (degradation of tables and indexes) influence the entire e-voting system at once and are difficult to repair.

The *risk of inaccessibility of validity confirmation/ time stamping service* is discussed Chapter 4 "Required and recommended security measures".

*Denial of service attacks*

It is not too unreasonable to claim that until the number of votes cast by e-voting does not exceed the number of conventional votes, malicious DoS attacks are not a serious problem for conducting the voting. The risk will not increase considerably even if e-voting were to become a prevailing voting method.

This opinion is based on the great incompatibility between the risk of the attacker and the motivation behind it. We do not believe that any political parties in Estonia would

dare to carry out a public, high profile attack against state information systems. There might be more motivation and audacity outside Estonia; however, it is much easier to block an external attack.
The length of e-voting – seven days – is sufficient for implementing measures against attacks originating in Estonia or abroad, which limits the temporal scope of such an attack.
*Impact of voter/ VA errors to the Central System*
A simple error committed by a voter /VA – such as sending a vote a hundred times in a row – can overload the whole Central System.

#### 9.4.4. VCA

VCA operability is critical only in the final phases of the voting. Since the VCA server does not contain dynamic data, it can be restored very quickly if a failure should occur. Therefore its technical operability is not an important risk.
*Failures of VCA application* – VCA application is so simple that the likelihood of random errors occurring in it is relatively low. The most likely location for problems is the communication between the application and the security module.
*Operability of VCA private keys* might cause problems. RSA encrypting operations run very slowly and this must be taken into account when choosing the private key hardware (security module).

#### 9.4.5. Auditing system and auditing application

Auditing application must run the functions of interim audit (log comparison etc.) allocated to it between the counting of e-votes and publication of election results. Since the volume of the logs is small, this should not constitute a problem.

### *9.5. Risks of key management*

#### 9.5.1. VCA private key management

The whole confidentiality of e-voting is based on the security of the secret key to the vote counting application.
If the *secret key is destroyed*, e-votes cannot be decrypted and the e-voting has failed.
If the *secret key cannot be accessed*, the counting of e-votes will be postponed. If the access has been cut permanently, it is equal to the destruction of the key.
The key could be destroyed/lost during the key management procedures, security module failures, chip card failures and problems with key managers, which range from sickness, forgetting the PIN-code and lack of time to being subjected to a targeted attack.
If a *secret key has been exposed*, the confidentiality of all votes cast has been violated. The relevant dangers have also been described in the conception.
Causes for exposure of the key include errors in carrying out key management procedures, conspiracy between key managers and security module failures.
The system must thus include measures for:
– ensuring the operability of the secret key,

– restricting the access to and use of the secret key.

### 9.5.2. VCA public key management

In reality the ensuring of the authenticity of a public key is a tougher task than securing the secret key. The weak point of all systems using public key cryptography is public key distribution, not private key protection.

Public keys are usually distributed on the basis of certificates enabling to derive trust for certificate owners and their public keys from the trust for one of the parties (certificate authority, CA). The analogy in the everyday world is the passport, whereby personal identification is based on trust for the state as the issuer of the personal identification document. The same pattern should be used when voting electronically.

*VCA public key substitution attack*

VCA public key used for encrypting a vote in the voter application ***must*** correspond to VCA private key. If the application contains a wrong key,

– the owner of the secret side of the new key can open the votes, thus exposing the votes;

– VCA can no longer open the votes and the votes go missing.

*Man-in-the-middle attack*, whereby the attacker encrypts the votes anew, this time with the right key, and forwards them by way of VCA to the voting system, is also possible. In standard elections, a comparative situation could be created by way of two-storey ballot boxes: the voters insert the envelopes into the upper compartment, but the falsified votes originating from the lower compartment are counted.

Digital signature requirement neutralises the risk of the man-in-the-middle attack – the attacker cannot imitate the digital signature of the voter and can thus intermediate only one vote (with his or her own signature).

*VCA key man-in-the-middle attack by VSS itself* might be a problem (web server gives the voter an application which encrypts the vote with a key known to VCA; VSS informs the attack organiser of the voter's choice, then changes the vote and encrypts it again), since this would allow the "attacker" to ignore the requirement to check the digital signature. This can be avoided through auditing, which checks whether there are valid digital signatures for votes sent to VCA in VSS. In reality, the web server has other, simpler ways of detecting and changing the voter's vote, such as modification of the VA.

*Attacks against key managers*

It is possible that attempts are made to influence or remove persons who can access the VCA private key, in order to ensure the failure of the e-voting or detect the value of the votes. The risk can be alleviated by employing a sufficient number of key managers, independent from one another in the organisational sense, and parcelling out access to keys among them (so-called many-of-many schemes).

## *9.6. Reliability risks*

E-voting differs from normal Internet-services by its political appeal. This process and system will almost certainly prompt formal protests and disputes, and we must take into account the attacks that create material for them.
Therefore we must be ready to face charges of unreliability of the system, or attempts to present it that way.
Our analysis does not reflect political risks, but we will describe the relevant technical possibilities.
*Complaint – inappropriateness of the system to public voting*
It can be said that e-voting does not fulfil certain requirements set to public voting – for example, it does not ensure the confidentiality of vote, correct result or uniformity of voting.
*Complaint – uncontrollability of the system*
One can claim that e-voting is resolved technically by constructing it as a secret, closed solution or that it is too complex for external observers to check.
*Attacking of public components of the system*
The public system components can be attacked with a lot of ado and high visibility. This involves DoS's, defacement of web server, modification of VA, etc.
The attack does not have to be directed against the structure of the system.
Defacement of NEC web server – e.g. by posting an indecent image on its front page – can bring along a lot of media attention without influencing the actual voting process in any way.
*Imitation of an attack, deceptions*
The general public can also be misled by a claim that an attack has been carried out, and falsified proof of it can be shown.
Anyone can add any pictures on the above mentioned NEC front page in their computer and send the result to the press, claiming that this was what the page looked like. Or claim that the VA did not function in their computer.
Similar attacks can of course be undertaken against standard elections: a voter can claim that his or her "passport was not asked in the station" and order a newspaper article on the subject.
*Errors of the auditing system and auditing application*
If the means for checking the system are faulty or insufficient, the quality of the system cannot be satisfactorily checked and its reliability will suffer.

# 10. ANNEX 4 – COMPREHENSIVE TABLE ON RISKS

We shall itemise the risks found in the form of a table and mark their probability and impact assessments. These are obviously very approximate in the case of a system in the conception phase.

Probability and impact assessments are marked on the scale of 1 – 3. Grade 1 probability means that "this will not happen anyway" and 3 – "this risk will almost certainly realise". In the "impact" column, 1 means that one voter or his or her vote is in risk, 2 – the risk is limited in time or in the amount of votes compromised, and 3 – will have an important impact on the voting result or the e-voting process as a whole. Risk assessments are given on the presumption that measures and conceptual improvements recommended in the present analysis are applied.

| **Risk** | **Location** | **Probability** | **Impact** |
|---|---|---|---|
| **Fundamental and process management problems** | | | |
| Risks arising from formalisation of processes | | | |
| Risks deriving from centralisation of processes | | | |
| Quality of system project – design errors | | | |
| Quality of system development – software errors | | | |
| System management quality –configuration and management errors | | | |
| Software problems arising from cryptography use | | | |
| Risks of voter's computer as an uncontrollable environment | | | |
| Risks connected to AIP use | | | |
| Possible conflicts of conventional and e-voting processes | | | |
| Limited confidentiality of e-voting results | | | |
| **Reliability risks** | | | |
| Complaint – inappropriateness of the system to public voting | | | |
| Complaint – uncontrollability of the system | | | |
| Attack against the public components of the system, e.g. defacement of web server | | | |
| Imitated attack and other deceptions | | | |
| Errors of the auditing system and auditing application | | | |
| **Risks influencing the correctness of voting results** | | | |
| Directing the voter to a falsified web page | net | xx | xx |
| Man-in-the-middle attacks between the web server and the voter application | net | x | xx |
| Attacking user's computer and gaining control | net | xx | xx |

| | | | |
|---|---|---|---|
| over it | | | |
| Functional failures of the voter application | VA | xxx | x |
| Use of misleading data by the voter | VA | xx | x |
| Intentional sendoff of an incorrect vote by the voter | VA | xx | x |
| Votes signed with an invalid certificate taken into account in the results | VA | x | xx |
| General risks of signature application | VA | xx | x |
| Unauthorised changing of input and output data of the system | Central System | x | xxx |
| Exposure of voter application | VFS, VA | x | xxx |
| Substitution of the VCA public key in voter application | VFS, VA, key management | x | xxx |
| Faulty candidate list/ unauthorised changing of candidate list | VFS, VA | x | xxx |
| Static content of the web server jeopardised | VFS | xx | xxx |
| Typical web application and web server errors | VFS | xxx | xx |
| Errors of input data in VSS | Data base | xx | xx |
| Unauthorised changing of input data in VSS | Data base | x | xx |
| Unauthorised changing of other data in the data base | Data base | x | xx |
| Changing the list of unsigned votes | Intranet | x | xxx |
| Functional errors of VSS applications | VSS | xx | xx |
| Digital signatures checking errors | VSS | x | xx |
| Functional failures of VCA | VCA | x | xxx |
| Discrimination errors | All components | xx | x |
| **Risks influencing the confidentiality of votes or voting result** | | | |
| Violation of the fact of voting in voter's computer | VA | xx | x |
| Violation of confidentiality of vote in voter's computer | VA | x | x |
| Violation of the fact of voting in the Internet | net | xxx | x |
| Violation of the fact of voting in the Central System | VFS | x | x |
| Violation of confidentiality of vote in web server | VFS | xx | xxx |
| Violation of confidentiality of vote in VSS | VSS | x | x |
| Leakage of the complete data base of votes | VFS, VSS, Intranet | x | xx |
| Leakage of information on e-voting usage intensity | net | xx | x |
| Exposure of VCA secret key | VCA, key management | x | xxx |
| Leakage of logs from the auditing system | audit | xx | xx |

| **Factors influencing the efficiency of the voting system** | | | |
|---|---|---|---|
| Errors in system management | Central System | xx | xx |
| Servers and communications network operability risks | Central System | x | xx |
| Failures and quality problems of Central System software | Central System | xx | xxx |
| Failures of Central System hardware | Central System | x | xx |
| Incorrect planning of necessary system resources | Central System | x | xx |
| Failures and quality problems of voter application | VA | xxx | x |
| Vast volume of the voter application (regarding the network connection of voters) | VA | xx | xx |
| Failures of the data base operability | VSS | x | xxx |
| Non-operability of validity confirmation/ time stamping service | VCS | x | xx |
| Service constraint attacks | net, VFS | x | xx |
| Central System overload caused by voter/ VA | VFS, VSS | x | x |
| VCA application failures | VCA | xx | xx |
| Destruction/ inaccessibility of VCA secret key | key management | x | xxx |
| Operability of VCA pirate key (decrypting speed) | VCA | xx | x |

# 11. ANNEX 5 – SECURITY MEASURES DEEMED UNNECESSARY

The working group discussed the following security measures in 2003 but did not set these as requirements. Their implementation in 2010 is naturally not prohibited.

**VA – VFS communication additional security features**

An additional security layer could be added to VA-VFS communications:

– VA could check whether the web page possesses the right certificate;

– a supplementary messages and data authenticity and integrity check could be used in addition to HTTPS: candidate lists could be previously digitally signed, etc.

Unfortunately this seemingly useful measure does not help against exposure of the application or man-in-the-middle attacks. Adding security on the application level does not protect against falsifying the application itself.

This method would raise the level of information needed for the attack (the application itself must be attacked instead of simply falsifying data), but would also further complicate VA.

**Division of VSS into two separate components**

It is possible to divide VSS into two simpler components: a server functioning during e-voting and later data processor. Only a data base would connect the two components.

This basically means the separation of data acquisition and data processing.

**Online paper-tally for guaranteed "recording" and auditing of incoming votes**

Immediate print-out of incoming votes to the Central System is possible. A paper trace of votes would thus be created. An alternative would be to print the fact of receiving a vote and the hash of the vote, basically it would be a paper copy of Log1.

Nearly all Western election machine analyses strongly advise to create a back-up paper trace.

Nevertheless, we think that it is possible to ensure the safeguard of votes and system auditing without resorting to antiquated technologies. The paper printout would only carry an artistic value.

**Local data bases in different components**

Keeping the voter data base in the web server (VFS) was also discussed.

Unfortunately this leads to such synchronisation problems that the possible added security would not compensate them in any way.

**Duplication of central system components and distribution of workload**

Proceeding from the system architecture, there can be an unlimited amount of central system components, except the data base.

This said, we see no reason to resort to duplication at the moment.

E-voting information system differs significantly from the "normal" e-service information system:

– its life is short – probability of physical failure minor;

– workload is low – there is no need for load-balancing;

– there are relatively few possibilities for testing and set-up.

In addition to the above-mentioned peculiarities, the e-elections that have taken place so far show that a great part of the load of users falls on the first and the last hour of e-voting period. Such a situation is characteristic of all systems where using is connected to a fixed term or time limit, like e-Tax Board. Still, it would not be reasonable to set the dimensions of the system by the rush hours. It has been tried to avoid this problem at e-voting by lengthening the voting period from three days to seven days.

This means that the probability of errors eliminated through duplication (hardware failures, overload) is very low, while the probability of management and software failures is very high. Component duplication increases the complexity and in the end reduces the serviceability instead of improving it.

**Duplicated RSA keys**

In the interests of operability, two different RSA key pairs could be used in VCA. VA includes both and sends votes encrypted with two different keys to VFS. This mechanism was also mentioned in the concept.

This is however not a good solution for two reasons.

– Integrity of the voting result is not guaranteed – we do not know if the two cryptograms include the same information. This creates a situation where we get a different result, depending on whether we decrypt it using the 1st or the 2nd key.

– We do not know if the 2nd vote was encrypted using the right key. Someone might have substituted one key in the VA with another and thus violate the security of all votes without the Central System detecting this.

The complexity of key management of duplicate keys makes it a feature best to be avoided.

**Response message via a third party (SMS: Thank you for voting!)**

Alleviates various abuse risks but creates even more. There is also no commonly used suitable channel for that.

**Authentication of central system network and users of operation system**

We discussed several ways for better authenticating the users of the Central System and zoning the network. The users could be engaged in a separate network and allow them access to servers only through a firewall. This would guarantee console logs, restriction of protocols, etc.

We reached the conclusion that it was not really possible or efficient to restrict access to system managers. In any case, certain activities can be carried out only in physical contact with the computer.

**Randomisation of VCA moving vote file**

This would have been an instrument for avoiding connecting votes to voters by VCA. This would practically mean implementation of mix-net structure inside the Central System.

**Voting intensity threshold values**

In case of a small number of e-votes cast (see chapter 9.3.7, “System output”) the value of the vote could be exposed through exclusion method.

This can be avoided by demanding that in case of low voting activity e-voting will be declared as failed.
We should define:

– the minimal number of votes required to count the result;
– the minimal number of accepted votes required to take the result into account.

We did however reach the conclusion that this was not necessary.

# 12. ANNEX 6 – REFERENCE WORKS

[BM] „Practical Security Analysis of E-voting Systems“, Ahto Buldas, Triin Mägi, IWSEC 2007
http://dx.doi.org/10.1007/978-3-540-75651-4_22
[Florida] The Butterfly Ballot: Anatomy of a Disaster
Ask Tog veebiajakiri, jaanuar 2001
http://www.asktog.com/columns/042ButterflyBallot.html
[IP1-S-EE] “Recommendation on legal and operational standards for e-enabled voting (Second Draft)"
IP1-S-EE töögrupp, juuli 2003
www.coe.int
[ISKE] Infosüsteemide kolmeastmelise etalonturbe süsteem ISKE,
Riigi Infosüsteemide Arenduskeskus, juuni 2010
http://ria.ee/iske
[Neumann] Security Criteria for Electronic Voting
Peter G. Neumann, september 1993
http://www.csl.sri.com/users/neumann/ncs93.html
[OWASP] The OWASP Top 10 for 2010
http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
[PKCS] PKCS #1 v2.1 RSA Cryptography Standard
RSA Laboratories, juuni 2002
http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/
[Rubin] Security Considerations for Remote Electronic Voting over the Internet
Avi Rubin, AT&T Labs – Research, juuli 2003
http://avirubin.com/e-voting.security.html