E-voting conception security:
analysis and measures

Authors:
Arne Ansper
Ahto Buldas
Mart Oruaas
Jaan Priisalu
Anto Veldre
Jan Willemsen
Kaur Virunurm

Document: EH-02-01
Date: 15.12.2003
Table of contents

1. INTRODUCTION

2. REQUIREMENTS AND PREREQUISITES OF E-VOTING PROCESS

2.1. PROBLEM OF CONFLICTING REQUIREMENT

2.2. SET REQUIREMENTS

2.3. TECHNICAL PREREQUISITES

2.4. ARCHITECTURAL COMPONENTS OF THE SYSTEM

3. IDENTIFIED RISKS

3.1. FUNDAMENTAL PROBLEMS

3.2. SUMMARY OF TECHNICAL RISK ANALYSIS

4. ADDITIONS AND CLARIFICATIONS TO THE SYSTEM

5. REQUIRED AND RECOMMENDED SECURITY MEASURES

5.1. GENERAL CENTRAL SYSTEM REQUIREMENTS

5.2. REQUIREMENTS FOR SYSTEM COMPONENTS

5.3. ORGANISATIONAL REQUIREMENTS FOR VOTING

5.4. KEY MANAGEMENT

5.5. SUMMARY OF MEASURES

5.6. RISKS THAT ARE TO ACCEPTED

6. GENERAL EVALUATION OF THE CONCEPTION

7. SUMMARY

8. ANNEX 1 – DATA PROCESSED IN THE SYSTEM

9. ANNEX 2 – DATA CHANNELS IN AND OUT OF THE SYSTEM

10. ANNEX 3 – TECHNICAL RISK ANALYSIS

10.1. RISK CLASSIFICATION

10.2. RISKS TO THE INTEGRITY
10.3. RISKS TO THE PRIVACY

10.4. RISKS TO THE OPERABILITY

10.5. RISKS OF KEY MANAGEMENT

10.6. RISKS TO THE RELIABILITY

11. ANNEX 4 – COMPREHENSIVE TABLE ON RISKS

12. ANNEX 5 – SECURITY MEASURES DEEMED UNNECESSARY

13. ANNEX 6 – REFERENCE WORKS
1. **INTRODUCTION**

The present 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 proposed by the National Electoral Committee. The task is to appraise the general suitability of security in the proposed solution, to thoroughly examine technical and organisational risks and to establish primary security requirements for the system in development.

Taking into account the relative simplicity of the voting scheme under, pinpointing the main risks is an intuitive process. The conception itself includes both a security analysis and protection measures against the major risks – digital signature, voice encrypting, and the division of Central System into several servers. The present analysis is more systematic; it discusses risks in greater depth and sets many more specific technical requirements. Yet at the moment we are merely dealing with a conception that cannot be subject to a very detailed analysis. It should, however, be ensured that security is paid attention to in the course of the whole development of technical solution.

Our work is confined to the processes of work security and operations. We do not appraise political risks or assess social or political aspects of electronic voting. Nevertheless, we do consider technical issues related to security that require to be decided upon or accepted 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 and reliability, and the necessity of technical auditing.
2. REQUIREMENTS AND PREREQUISITES OF E-VOTING PROCESS

2.1. Problem of conflicting requirements

The conflict between confidentiality and controllability introduces complications into secret voting. On the one hand the attainment of correct results has to be ensured; in order to do so the whole process needs to be subjected to auditing at all stages, every action needs to be traced. One the other hand, the confidentiality of votes has to be secured to keep the process democratic; that is why it should be secured that no link is established 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. 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 set for the system have to be a matter of conscious choice and an agreement reached at a political level. Auditability and secrecy, error protection and unprovability, security and comfort do not go hand in hand; a line should be drawn and a decision should be made. Nearly all e-voting security studies discuss this contradiction underling the task. The best analysis made so far is apparently that produced by Peter Neumann [Neumann].

2.2. Set requirements

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

2.2.1. Correctness requirement of voting

Correctness or integrity of voting includes functional requirements the fulfilment of which secures correct result of the voting reflecting the choice of voters and being 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. Below we set out only the most important security requirements.

Authorisation of voters – only voters whose names are mentioned 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” – only one vote has to be considered of all votes given by the voter regardless of the way they 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 the two functional requirements that e-voting systems are often subjected to but which are not accepted in the Estonian law and that the technical solution of e-voting does not directly support. In fact both of them are realised on the basis of the principle of supremacy of conventional voting.

Annulability 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 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.

2.2.2. Secrecy requirement of voting

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 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 can trace the connection to the web server of the National Electoral Committee; observers spot those walking into the polling division. At the same time no-one 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 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 the way he 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 results of e-voting are not public.

2.2.3. Operating reliability requirement of the voting

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 fast enough, and ensure the preservation of data and timely presentation of voting results.

2.2.4. Reliability requirement of voting

The society and those concerned 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 have to have the opportunity to ensure 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 process of calculation.

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 bring two requirements set for the record 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 secrecy (file-safe) of votes, or in other words the requirement that under no circumstances should a voter’s vote become public including the situation where an agreement exists between all the other parties (e.g. those responsible for the organisation of voting). We believe that there is no and there cannot exist in reality a voting scheme meeting such requirements.

2.3. Technical prerequisites

Technical prerequisites underlying the presented conception and the current analysis are as follows:

Central System servers are secure and reliable. That means that compromising a Central System server may have an effect on the security of e-voting to the extent 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 operatable and its physical security is in order. The analysis does not include intermediation attacks, connection and power failure interrupts.

The information system of e-voting is separated at the network level from the rest of the voting information system, the connection with the outside world is established by means of limited interfaces.

Voters have an ID card provided with valid authentication or digital signature certificate. Digital signature is unfalsifiable, the ID card and its software are secure and error-free.

Voter’s environment (computer, browser) is not controllable and risks related to that can only be addressed by communicating and raising 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 conception. 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 conception.

Voter’s application, VA – application which encrypts the name of the voter in the 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 equally referred to as the Web server.

**Vote storage server, VSS** – server which stores the votes given by the voters and enables to sort, delete and forward them to the VCA.

**Vote counting application, VCA** – a separate application which sums up digitally unsigned e-votes and delivers the results of the e-voting. The computer running the VCA is called VCA server.

**Time-stamping service** – an external service confirming the time when digital signatures were given.

**Internet** – network connection between the voter’s application and the Central System.

**Intranet** – connections between components of the Central System. **Firewall** and other possible mechanisms of access control at the network level along with offline data carrier between VSS and VCA equally belong to the intranet.

**Audit system** – component of the Central System dealing with gathering audit data and working with audit application.

Furthermore, 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 a separate logical component and could be placed on a separate server.

The **general elections information system** which generates data required for the e-voting, where annulments and restorations are directed from and where e-voting results 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 merely deals with conceptional 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 PC has nowadays reached such a level that voters believe they are dealing with a *black box* they cannot nor are able to control. The PC can do virtually everything on behalf of the voter but behind his back – vote for a different 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 PC can be exposed to attacks:
- network / operation system (e.g. Microsoft RPC errors),
- e-mail as the most common open Internet-based service (e-mail viruses),
- web browser security errors (every browser contains an amount of errors sufficient for taking a computer over),
- physical access.

Using these channels one can install software which either
- traces user’s actions – gets to know his vote and/or the PIN code of his or her ID card; or
- replaces the voter’s application with a different one and gives a wrong vote (different from that of the user); or
- abuses the ID card and gives digital signature of the voter who is not aware of the fact; or
- blocks the vote.

Avi Rubin’s work discusses the problems related to the computer of the voter in greater detail [Rubin]. Unfortunately these are risks that the Central System of e-voting can neither control nor avoid.

Still we believe that they are acceptable.

The reasons for that are as follows:
- There is no sense in attacking one voter’s computer or network in order to falsify his or her vote. Only a massive attack can have an effect on the result of the voting.
- Attacking a voter’s computer with the aim of acquiring his passwords/PIN codes is not related to e-voting, this can equally be done at any other time.
- A secret massive attack on PCs is practically impossible. This has been demonstrated by the way viruses have been spread - regardless of their artfulness and stealthiness they would manifest themselves in computers of certain kinds and often because of computer’s own errors (those of Windows, browser, Outlook etc.). 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 blocked.

This can cause the failure of e-voting but in all probability the attacker will be captured. This implies that one cannot make a political order to conduct such an attack.
- It is possible to conduct a massive falsification attack on server if there is no active communication with voters from the web server’s address.
Given the fact that voting takes place using a non-standard technology in web terms (a separate application), the falsification of votes requires a considerable knowledge. It is much easier to attack an internet-bank and the profit gained from it is considerably more realistic.

**Risks related to AIP**

Computers used by many people should be considered separately: AIPs and cyber cafés, school computer classes, centrally managed computers of large companies. Administrators of such computers and networks have the possibility to attack all voters using these computers for instance in order to make them vote 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.

**3.1.2. 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. It will not be possible to establish limits to further actions if someone succeeds in making the voter begin voting from a wrong page: he can be sent a wrong application, given a wrong list of candidates and his whole computer can be taken under control. These risks are analysed in greater detail in the risk analysis section 0, "The risk of discrimination 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.

".

**3.1.3. 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 conception – consists of components that we simply have to trust. Relying on trust can be reduced to a minimum by solely fetching those components from reliable sources that should thereby not be aware of the fact that they are planned to be used. However, the fundamental problem remains to be solved.

**3.1.4. 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. Voter’s application will certainly not work on all computers citizens of the Republic of Estonia have access to. 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.

**3.1.5. Possible conflicts of conventional and e-voting processes**

E-voting takes place at the same time as conventional voting; apparently the same persons 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 and divide their attention between several systems.

The conflict of polling divisions’ priority can be brought as an example. In order to start counting e-votes all annulment applications have to have come in from all the polling divisions, these have to be transferred to the National Electoral Committee by electoral
district committees, whereupon they reach the Central System. A delay in the work of at least 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 that and there is no compulsion to do so.

3.1.6. 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 failure by a person counting votes to provide a fair result can have a considerably greater effect than that of conventional voting based on detached process.

3.1.7. 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 feasible ways of making exceptions that have worked so far.

The example of voter’s identification can be brought 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; they place reliance upon the spouse, 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 $\leftrightarrow$ VSS protocol contains no "blandishing" messages.

A more drastic example is the automation of London ambulance dispatch. The previously informal management of ambulance transport was terminated and as a consequence 26 people died in three days not getting any help.

It is also necessary to consider the election procedures where tendencies exist at present to break the rules and reflect whether problems can arise in similar e-voting procedures. An input must come here from the lowest possible levels of conventional voting system organisation.

3.1.8. 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 critical. 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.

3.1.9. Development and management problems

Two highest risks of every information system is the quality of development or software faults and the quality of management or system configuration faults. Along with errors untested software or negligent management can 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 servers and the voter’s application, operations 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 in the first place to be expected in the operating reliability of the voter’s application (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 to develop PKI applications than software having similar functional complexity but not relying on cryptography.

3.2. Summary of technical risk analysis

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

– Risks deriving from the Internet as an open and hacker-friendly environment;
– Voting process errors, magnified by the fact of working in an unknown environment;
– 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 given in "Annex 3 – Technical risk analysis", the following is a list of what we consider the ten greatest risks (with the strongest impact and probability).

<table>
<thead>
<tr>
<th>Risk</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack against user’s computer and gaining control over it</td>
<td>VA</td>
</tr>
<tr>
<td>Failures and quality problems of voter application</td>
<td>VA</td>
</tr>
<tr>
<td>Man-in-the-middle attacks against web server and voter’s computer, fake web pages</td>
<td>Internet:</td>
</tr>
<tr>
<td>Exposure of voter application or its input data</td>
<td>VA, VHS</td>
</tr>
<tr>
<td>Defacement of web server or unauthorised changing of its contents</td>
<td>VFS</td>
</tr>
<tr>
<td>Violation of vote confidentiality during voting in web server</td>
<td>VFS</td>
</tr>
<tr>
<td>Traditional web application/ web server management and security errors</td>
<td>VFS</td>
</tr>
<tr>
<td>Failures and quality problems of Central System (VSS) software</td>
<td>VSS</td>
</tr>
<tr>
<td>Functional failures of VCA</td>
<td>VCA</td>
</tr>
<tr>
<td>Destruction/ inaccessibility of VCA secret key</td>
<td>Key management</td>
</tr>
</tbody>
</table>

The biggest security risk is the security of web server contents and applications. Here, the great probability of risks and easy execution of attacks combine to influence the whole voting process. Central web servers of the Estonian state (*.riik.ee) are notoriously filled with loopholes. In fact, it is difficult to find any well secured servers from the Internet.

We identified no formerly unacknowledged fundamental problems that have not been taken into account in the conception. The simplicity of the scheme means that errors are also simple and the intuitive result is not much worse than technically systematic risk analysis.

To summarise: risks of e-voting are in fact very similar to the risks of conventional voting, most technical attacks and threats have analogies in the material world. IT-systems are replaced by people and organisations, but schemes and processes remain the same. Instead of voter application errors, election lists could be wrong/faulty („Florida butterfly effect”), software and people alike make mistakes in checking and counting the votes, problems arise with operability (queues form at polling stations) and reliability is contested (protests).
E-voting only adds dependency on technical equipment into the equation. In addition, the use of technical equipment serves to seemingly magnify the general problems: frequency of errors is reduced but their scope extended.
4. **Additions and clarifications to the conception**

The architecture of the system was discussed in detail in the course of analysis and, of course, certain parts were identified where the introduction of a different solution than that proposed in the initial document was considered necessary. The proposals made are as follows.

It should be mentioned that the matter concerns rather technical details that do not introduce any considerable modifications into the voting scheme.

4.1.1. **Verification report of the voter’s application and modification of the data communication scheme**

We propose to modify the voter’s application in such a way that the verification report concerning the choice of the voter would be given by the voter’s application upon the verification of candidate’s number and name.

This ensures security in cases where voter receives misleading information about candidates from some source. The possibilities for that are laid out in the risk analysis - an external attacker having control of the browser display image, sending a misleading e-mail recommending to vote for a candidate with a given number etc. In this case a voter’s application originating from an authentic source is the most proper and secure place for verification report. The web server should not do the confirmation since this would be in breach of the vote secrecy requirement.

The voter’s application should thus receive or contain information on possible candidates.

There are several options for receiving it. These include:

- After having authenticated the voter the web server transfers to the voter’s application possible lists of candidates (names and numbers only) for the specific voter. The web server displays further information on candidates while the voter’s application solely poses the control question.

- After having authenticated the voter the voter’s application receives all the necessary information from the web server for the specific voter. The application displays candidate information, not the browser. The application also poses a verification question.

- The whole list of candidates (candidate choices) is integrated in the voter’s application. The information is displayed by the voter’s application itself and not the browser.

It is not essential from the security point of view which of the options is chosen. It can also be that the choice made varies from one voting to another – this depends on the volume of data on candidates/choices and the complexity of information used. The latter in their turn depend on specific elections – data volume differs approximately 10 000 times in cases of referendum and local elections.

The introduction of additional security measures between the voter’s application and the Central System has equally been under discussion. It is possible to introduce further authentication and message signatures (list of candidates, reports). However, these measures were found to be inefficient – channel security **has to be** ensured at the browser – web server level.

As a result we are altering the viewpoint underlying the conception that the voter’s application is merely an encrypting mechanism that does not contain any information regarding the e-vote and that it communicates with the Central System solely to transfer the vote.
4.1.2. Storing candidates database in the Central System database

According to the conception during e-voting the list of voters is kept outside the intranet of the Central System. VFS and VSS address queries to an external information system (AS Andmevara).

This means that the e-voting information system depends on yet another external party, its availability and the speed of service provision. (In order to find the district) the list of voters has to be checked at the moment of authenticating a voter, that is queries are time-critical.

We believe that it is more reasonable to produce a copy of such database for the e-voting.

**Voters’ database needs to be located within the Central System network.**

Because the list of voters can change during e-voting, the problem of updating the database (synchronising with the AS Andmevara voters’ database) needs to be solved. There exist many possible solutions – regular requests for update from the Central System, event-based posting of updates by AS Andmevara or making additional requests to the AS Andmevara system in case a voter wishing to make a vote is not on the list in the Central System.

The most reasonable option is the regular updating whereby the Central System of e-voting requests the latest updates from the AS Andmevara database. This process causes a small delay while the registering information of a new voter is transferred to the e-voting system, but the delay can be reduced to absolute minimal, which causes no actual problems. The update will have to be realised in the so-called pull technologies: the database of the e-voting system requests updates from the external data source (AS Andmevara). The external data source will certainly have to be authenticated.

Producing copies and regular updating is the better solution also from the AS Andmevara database point of view than checking every voter proposed in the conception because the exact frequency of requests and the load cause by them is known.

4.1.3. Storing the list of candidates in the VFS and VCA server

**Static list of candidates has to be statically located on the discs of VFS and VCA servers.**

It does not need to be stored in the Central database.

VFS needs to have access to the list(s) of candidates in order to transfer them to voters. As we have described above, the VFS application has to be as simple as possible and, if possible, contain only static information. Reading the lists of divisions from the database and formatting them for the web browser and voter’s application is the task that can and needs to be completed before the beginning of the e-voting.

VSS requires no access to the list of candidates.

VCA that does not have network connection requires a copy of the list in any case.

It should be possible in some way to control the integrity of file(s) containing the lists of candidates (checksums etc.).

4.1.4. Connecting voter’s authentication and digital signature

While giving a vote the voter is twice subjected to authentication:
- The web server authenticates the voter by means of ID card in order to identify his or her electoral district;
- VSS identifies the voter by means of a digital signature.

These two ways of authenticating have to be connected to prevent intermediation attacks.

**The user authenticated in the web server has to be the same person as that signing the vote.**
If it is not the same person, an attack is probable. The vote should then not be accepted, the voter has to be sent an error report and the application has to log the fault as a possible attack.

The aim of this measure is to prevent intermediation attacks. An intermediation server that interferes with the communication between the user and the voter of the right server can be created for every web service. In case of e-voting such intermediation server can serve false application, wrong candidate date, block the votes of some voters – Cf. section 0 “The risk of discrimination 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.”

Conventional method or the authentication of the web server (SSL certificates) provides no ultimate guarantee against it – an intermediation attack is possible and an alert and attentive voter can simply detect it as a browser’s error report or the non-use of HTTPS protocol. An intermediation attack can also be conducted with the voter being aware of it. For example, it can be used to create a server for buying/selling votes.

Two-way SSL authentication excludes intermediation attack at the technical level – we can be certain that no-one stands between the authenticated voter and the VFS. For this certainty to be of use one should also know that the authenticated voter is in fact the giver of the vote – in other words it should be checked that the vote sent in the course of the SSL session has been signed by the person who initiated the session. Otherwise the author of the intermediation attack can authenticate himself or herself to the web server with the help of his or her ID card and forward the intermediated vote accompanied by the right digital signatures.

This method equally fails to provide absolute protection against the risk of accessing the wrong web server. The voter’s computer which accessed the wrong web server can be completely taken over. However, in this case we are not dealing with an intermediation attack but with one of the ways of attacking the computer of the voter tackled in point 3.1.

4.1.5. Checking digital signature validity

Checking validity of digital signatures is crucial to prevent voting with outdated certificates (e.g. stolen ID cards), the latter being both a fault in essence and a direct violation of law. But in what way and at what point should this be checked?

With this respect the conception mentions the necessity to obtain a validity confirmation for each digital signature. This would enable to obtain a confirmation from the issuer of the certificate that the signature is valid along with the certificate. Moreover the signature would inevitably be linked to the moment it was given.

This solution gives rise to two problems:
- **operability risk** – the process of voting depends on the external reliability of the party (issuer of the validity certificate) and the network connection with it;
- **secrecy risk** – the issuer of validity certificate would obtain the list of all e-voters.

Both risks can be formally reduced by means of contracts: a confidentiality contract and a contract securing operating reliability of the service can be entered into with the issuer of validity confirmation requiring the provider of service to ensure its availability and speed and not to disclose data to anyone. Yet, contracts are a **prohibitive measure** rather than an instrument of preventing risks. It is not reasonable to use contracts where a secure solution could simply help avoid the problem.

Having discussed the matter at length we recommend the following alternative.

The validity of the digital signature shall be checked by VSS itself using the Certificate Revocation List (CRL) obtained from the provider of certification service. Taking into account the fact that lists are created with a certain reference (twice per day in the case of AS...
Certification Centre) the risk exists that the revocated certificate will not be reflected in the CRL of the VSS. To prevent this time stamps should be established for digital signatures and at a later point, for instance at the time of vote sorting signatures should be checked again using the final and at that point sufficiently recent revocation list. In this way signature and voters’ information will be kept within the Central System and the certificate validity check will be secured.

The following risks remain in case this solution is applied:

- dependence on the operating reliability of the time stamping service provider (TSP);
- synchronisation of the TSP time with that of CRL generator;
- the fact that the VSS will temporarily accept votes with invalid digital signature;
- the risk of giving the voter the wrong feedback about counting such vote.

However:

- There exist several TSP in Estonia and any of these can be used – should there be a failure it is possible to switch from one service provider to another;
- it is possible to time-stamp a vote offline, since no negative response is given to time-stamping and this does not have an effect on the feedback given to the voter with regards to counting his or her vote;
- the risk of the wrong feedback to the voter is minimal. It will only occur when the vote is given with a recently suspended/revocated certificate and this information has not yet reached the Central System. Legitimate voter cannot reach such result; this will either be the case of an attacker (who uses a stolen ID card) or a voter making fun at his or her own expense.

Our time-stamping / revocation list recommendation is not absolute. If the mechanism of checking the validity of signatures based on validity confirmations is considered significantly preferable it can be used provided it is covered by the abovementioned contracts.

4.1.6. Vote verifiability function in voter’s application

The conception puts forward two ways of satisfying the requirement of vote counting verifiability: audit application and an automatic web application.

The easiest way is supplying voter’s application with this function which would identify at the beginning of the voting process whether the voter has already given vote and would give a corresponding notification. This can be realised in the form of a separately launched (non-automatic) request.

Upon completion of the e-voting the VFS and the VSS have to block the function of sending votes and retain the function of verifiability of votes.

The audit application should exist regardless of this.
5. **REQUIRED AND RECOMMENDED SECURITY MEASURES**

We will not waste room on standard requirements for constructing secure systems – these can be found in any books or standards on data security. We presume that access control is applied, management activities are documented, operating systems (the „black boxes” of the conception) updated in the field of security errors, applications check their input and log in the completed activities, and so forth.

*We recommend that the general security of the system be based on Information Systems Three-level security model for information systems (ISKE) High security class.*

At the same time no ultra-secure system can be secured by reference security alone. We will therefore detail the security requirements deriving from specific features of the e-voting system.

5.1. **General Central System requirements**

5.1.1. **Central System architectural requirements**

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

**Separation of the Central System**

E-voting Central System must be an 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 absolutely minimal necessary for providing required services and running applications. Servers should not contain development means (compilators, support of superfluous programming languages), data base access, etc. Unnecessary applications and services must not be installed, only defined network protocols should be open.

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

The system must detect in real time the network level attacks (IDS on network level) committed against it. Servers must detect violation of integrity of files important for the operating system and e-voting. Some tripwire-type mechanism can be used, among others.

**Recording of server 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 a functioning configuration as soon as possible.

After the end of the voting period, a „frozen” auditing copy must be made of all VFS and VSS hard disk servers and the data contained in these.

A second auditing copy must be saved just before the final establishing of the results.

Auditing copies must be stored securely – in a security envelope and locked safe – and their use 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). Cf. 4.1.2. VCA constitutes an obvious exception because it requires all data to be submitted as static files.

5.1.2. Central System applications requirements

Non-graphicality of user interfaces
Since all operation carried out are extremely simple, we recommend that only applications of textual or pseudo-graphical interface be used in the entirety of the Central System. This enables the use of simpler development tools, improvement of application transparency, means that graphical interfaces no longer need to be installed in the Central System servers, etc. The negative side is the Spartan (more technical, less attractive) appearance of the application. In addition, their use must be documented in detail, which 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 occurred during their work.

5.1.3. Ensuring reliability
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. The time-frame for responding to a voter, sorting and counting votes, etc, should be determined. 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. The results must be recorded. System manager must be notified of any errors.

Rapid server restorability
Before the beginning of the e-voting, so-called „clean” copies must be made of the 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.

Restriction of data loss, data restorability
The amount of data possibly lost through Central System errors must be limited. There are two methods for this – duplication and possible repetition of all input data, or mirroring of data to another system via software.

The simpler method is to make frequent back-up copies of data generated during e-voting (of data base redo-logs). Due to the small data volume, 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.)
5.1.4. Data format requirements

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

Measures should be implemented when transporting data between VSS – VFS and VFS – VCA 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, such as calculating checksums, should be implemented during transport.

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.

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 – ASCII text is the best.

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. Even though 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 forbidding e-voting. It is clearly more comfortable to vote using an ID-card, since this combines authentication and signature functions. This is however not a requirement – if other digital signature certificates exist during voting, these can be used.

5.1.5. Requirements of external data channels

Accessibility of candidate lists

Everyone who so wishes must have access to 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 originals. Thus, there could be a request in the Central System and NEC data bases which calculates the checksum over permitted voters’ personal identification codes.

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 authorised persons.

5.2. Requirements for system components

5.2.1. Requirements for voter application

**Independent verification report**

After choosing the candidate but before signing the e-vote, the application must display to the voter the candidate's name and number (and possibly also the political party) and ask the voter once more if he or she really wishes to vote for that candidate. The application should receive information for that elsewhere, not from the data entered by the voter. For reasons see section 0, "Verification report of voter's application and changing data communication scheme".

**ID card authentication certificate** should be used to authenticate the voter.

The application must not buffer the access codes of the voter's ID card digital signature certificate.

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

Viewing the data on candidates in the browser / HR 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.

5.2.2. Requirements for VFS / Web server

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

Communication between the web server and the voter's computer / voter application 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 equal to voting in a party headquarters instead of a polling division: nothing can be guaranteed, the result has no connection with the will of the voter. Additional explanations about such man-in-the-middle attacks are in section 0, "Connecting authentication of the voter and digital signature of the vote".

The server must work in a secured system, i.e. use HTTPS protocol.

The voter can check the authentity 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 do 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 work but also be installed!) in the web server.

**Authentication of the voter with ID card authentication certificate**

5.2.3. The voter must be authenticated with the ID card 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 reason is given in section Connecting voter’s authentication and digital signature

5.2.4. Connecting voter’s authentication and digital signature

5.2.5. Connecting voter’s authentication and digital signature

5.2.6. Connecting voter’s authentication and digital signature

5.2.7. Connecting voter’s authentication and digital signature

5.2.8. Connecting voter’s authentication and digital signature

5.2.9. Connecting voter’s authentication and digital signature

5.2.10. Connecting voter’s authentication and digital signature

5.2.11. Connecting voter’s authentication and digital signature

5.2.12. Connecting voter’s authentication and digital signature

"Connecting authentication of the voter and digital signature of the vote".

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.

For e-voting we recommend to reserve separate server name (FQDN) that is used only for e-voting.

Restricting the data on candidates displayed by the web server

On e-voting page or voter's application there may be only such information on the candidate that has been officially deemed necessary and is uniform for all. 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 application 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 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 works without active scripts. Web pages should be validated following the simplest (earliest) HTML standard possible.

Logging of correct votes

VHS 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 VHS 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 technical faults

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

Confirmation of acceptance/rejection of vote

VFS must send to the voter's application 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 conception (taking part in conventional voting etc.).

Identification of attacks at application level

It is advisable 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.
5.2.13. 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.

Requirements for verification of digital signatures

New mechanism for verification of digital signatures, that is different from the mechanism in the conception, is described in section 0, "Verification of digital signatures". With such scheme, the main requirement is the obligation to verify itself, there are no additional safety requirements. Wholeness is guaranteed by the mechanism itself, operability by the possibility to work in moveable regime and multitude of TSPs, confidentiality by exchanging only hashes (not certificate data) between the Central System and TSP.

Error-freeness 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 conception. 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 is planning to cancel as many votes (and also randomly - the same votes) as there are in NEC annulment list.
5.2.14. 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 physical data carriers (CD, floppy disk, printer paper).

VCA input and output to plain text

All VCA input and output, incl. the list of candidates, must be in plain text (for example, 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 operative 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.

VCA server must not have swap file (buffering of virtual memory on hard disk).

After counting and export of votes the VCA server should be booted.

Vote format controls

Before all other controls, type and format control must be carried out on decoded vote. Logical controls described in the conception (does the candidate with such number exist etc.) can be carried out only after that. The reason 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 look whether there is actually the number of the candidate or Unix shell script in the cryptogram.

Printout of voting results directly from VCA

We recommend to print out the voting result directly from VCA immediately after counting the votes and sign it by all key managers. This is the so-called "original" of the voting result.

5.2.15. 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 system servers' users log-ins (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.

5.2.16. Requirements for the management process of the system

Model security measures 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, not using the server configuration from previous voting.

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 advisable. A copy of them with limited data (only digital signatures of votes) and functions could be made 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 in use after 30 years. Thus all votes given must be deleted from all data carriers or these data mediums must be destroyed after final confirmation of election results.

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 voter application.

5.3. Organisational requirements for voting

5.3.1. 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 those election procedures where at present there is a tendency to break the rules and think whether e-voting will cause problems in these places or not.

Immediate forwarding of e-voting annulment lists at polling divisions must be motivated in some way.

5.3.2. 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 contain:
- 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.

Appointing executive and responsible parties

NEC must appoint the persons who execute 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;
E-voting conception security

- 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 special 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.

5.3.3. Requirements for the publication of system documentation

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

Conception 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 getting the full list.

Protocols used in public network (essentially 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 that are determined by NEC.

Informing voters of 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, 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.

5.3.4. Quality of service agreements

E-voting system manager must sign quality of service agreements:
- with the provider 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;
- if necessary, with the providers of time stamping service on the operability of time stamping service.

In addition to that there must be at least informal agreements with larger ISPs.
5.3.5. Security control during the system development

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

5.3.6. Pre-voting expert opinion on security

Before e-voting it must be evaluated whether the technical environment is sufficiently secure for e-voting. 2003. The second half of 2003 has vividly shown how rapidly the security of the Internet may decline. In the case of serious security violations of some of the technologies used, e-voting should be cancelled or, if necessary, the security loophole fixed (for example, firewall platform replaced with another).

5.3.7. 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 fix that no gross, easily detectable security violations took place during voting.

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.
- Cast an eye on IDS logs.
- Check the wholeness 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.
- 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 counts.

Interim audit must be done by technically competent persons who
- are not connected with the counting process of election results;
- 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.

5.3.8. Post-election audit

The main function of the audit is to check whether all voting activities 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, to change either the system or its security measures for the next elections.

5.4. 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.

Starting the vote counting application is the high point of e-voting. 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. Therefore it is necessary to be careful and thorough in describing and following these procedures, and what is the most important – to understand every moment what is being done.

### 5.4.1. 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:** Voter application 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.

### 5.4.2. 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 managers would be too great. Or: Several key managers must be present for using the key, but all of them do not have to be present.

Traditionally key managers must be independent, i.e. belong to different organisations. A representative (or representatives) of NEC must certainly be among the key managers but the key must not be used with the help of NEC representatives only.

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

A written act signed by all key managers 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. In longer time perspective, shorter key may not be safe, 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.

**5.4.3. Schemes 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 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 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.

**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 different key manager. Thus there are altogether N×K chip cards and key managers 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 has a different key, there are several copies of each key; one key for each lock is needed to open the door.

**5.4.4. 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 safety model. Therefore the procedures presented are general, offering one possibility for meeting the requirements defined above.

**Creating VCA key pair**

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

- Key managers authorise security module, using chip cards and PIN codes.
- Key managers give security model an order to create VCA key pair.
- Security module generates private key and public key.
E-voting conception security

- 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 safety 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 console of the security module.
- 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 model an order to export VCA private key.
- The security module exports the key by components to chip cards.
- 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 model 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 read private key.
- Security module is brought into ordinary, unauthorised regime.
- Security module prints out the calculated public key. The printer used is connected to the safety 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 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 voter application whether the whole process works and if VA contains the right public key. For that, one or more votes must be given with the help of VA and checked if they can be opened by VCA and is the result 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 with 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 model 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. file transport 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 with 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 model 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 content of results file, Lg4 and Log5 is 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.

For that, the static memory of security module(s) must be deleted. Relevant procedures are given in security model instruction. The producer of security model is responsible that the keys situated in the security model are irretrievably destroyed.

**Integration of VCA public key (certificate) into voter application**

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

After the final completion of voter application 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.

**5.5. Summary of measures**

Let us sum in brief all presented security measures.

**5.5.1. Technical security measures**

For guaranteeing general security we recommend to use the measures of the 'High' security class of the three-level security model for information systems (*Infosüsteemide Kolmeastmeline Etalonturbe Süsteem, ISKE*).

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 conception offered here are realised, neither updating voters’ database nor getting time stamps are no longer time-critical. In all components the purpose of design is simplicity, restricting of qualities 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 fault-freeness of software and correctness of management than on functionality and capacity. The aim is not fighting against denial-of-service attacks.

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 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 most vulnerable server. Key words are again simplicity, restricting of qualities 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 enable irregular 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.
5.5.2. 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 state practice of the 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 complemented, not disturbed, one another. The procedures must be described and tested, at the same time the danger of over-formalising 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 work during the voting period.

5.6. Risks that are to be accepted

5.6.1. 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.

5.6.2. Need to trust voter's computer and public network

We think that the possible insecurity of the voter's PC and Internet connection 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 intrusion do not have a motivation for that; and the political forces who have the motivation cannot take the risks connected with such 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 this simply has to be accepted.

5.6.3. Need to trust Central System computers

The fact that the components of the systematic layer of Central System computers simply have to be trusted must be accepted. Obtaining these components from trustworthy sources diminishes the risk.
5.6.4. Impossibility to support all voters
The fact that only the users of more widespread personal computers can vote must be accepted.

We do not make any recommendations on the platforms supported by concrete applications. Essentially the choice is between Java and ActiveX technologies. Software producers making applications for ID card have faced this choice also before, there is no solution that suits all and unfortunately the decision must be a compromise between politics and technology.

5.6.5. 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.

5.6.6. Risks deriving from formalisation of processes
The rules of physical world are always "soft" because relations between human beings can always be changed. In information system, however, the rules become strict and it is no longer possible to ignore them or "cut corners".

As a result of that, over-formalised 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.
6. **GENERAL EVALUATION OF THE CONCEPTION**

**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.

The analysed conception has done that and in our opinion reasonably.

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

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Way of guaranteeing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorisation and authentication of voters</td>
<td>Guaranteed with system design. ID card is stronger way of authentication than showing a paper document.</td>
</tr>
<tr>
<td>&quot;One person – one vote&quot;</td>
<td>Guaranteed with system design.</td>
</tr>
<tr>
<td>Prohibition of falsification of votes</td>
<td>Guaranteed with system design and auditing. Digital signature is unfalsifiable, other errors are identified during interim audit.</td>
</tr>
<tr>
<td>Uniformity of voting</td>
<td>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.</td>
</tr>
<tr>
<td>Possibility for electronic re-vote</td>
<td>Guaranteed with system design.</td>
</tr>
<tr>
<td>Supremacy of conventional voting</td>
<td>Guaranteed with system design.</td>
</tr>
<tr>
<td>Annulability of vote by the voter</td>
<td>is not a requirement but is guaranteed by the possibility to change one's vote</td>
</tr>
<tr>
<td>Possibility to give an empty vote</td>
<td>not a requirement, not guaranteed</td>
</tr>
<tr>
<td>Secrecy of vote</td>
<td>Guaranteed with strong encrypting and key management</td>
</tr>
<tr>
<td>Privacy of the fact of voting</td>
<td>Guaranteed softly – monitoring network connections by their providers is possible</td>
</tr>
<tr>
<td>Unprovability of voting</td>
<td>Guaranteed with the qualities of VFS/VSS applications and the possibility to change one's vote by conventional voting</td>
</tr>
<tr>
<td>Operability of the voting system</td>
<td>The scheme is modular and as simple as possible. Technical operability is no problem, software failures should be avoided by testing.</td>
</tr>
<tr>
<td>Transparency</td>
<td>Guaranteed by the simplicity of design, publicity of system principles and controllability of source code of applications</td>
</tr>
<tr>
<td>Auditability</td>
<td>Guaranteed with system design. Technical realisation through logging, audit system and audit application</td>
</tr>
<tr>
<td>Controllability of calculation</td>
<td>Guaranteed with voter feedback possibility and audit</td>
</tr>
</tbody>
</table>
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 discussion it was decided to discard them or use them only as abstraction 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.

Realisability

We believe that the technical side of the conception is realisable in Estonia with the help of currently existing IT knowledge. Limited number of known technologies is used, no insoluble problems can be foreseen.

Compatibility with the European Union recommendations

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

Solving the controversial problem of electronic voting is an interesting challenge. A simple solution will not allow meeting contradictory requirements. 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 eventually reduces security. There has to be an optimum in the choice between the theoretical security of voting scheme and the complexity of its realisation, and the technical solution corresponding to it gives the best compromise between the requirements set.

The analysed conception offers one, comparatively simple voting scheme.

Its strong points are:
- comprehensibility and similarity to conventional voting;
- maximal use of the security solution existing in Estonia (ID card);
- using only simple encrypting algorithms,

and last, but not least
- apparently it can be realised with the help of IT knowledge existing in Estonia at present.

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 scheme can be 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, i.e. division of tasks and responsibility, formal procedures, awareness and managing of risks by NEC, prepared action plans for solving emergency situations, independent audit, have to be added in accordance with the technical measures (cryptography, intrusion detection, double control of data etc.).

We believe that on the basis of the conception offered by us it is possible to create an e-voting mechanism whose security is higher than that of conventional voting with ballot papers. This 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.
8. ANNEX 1 – DATA PROCESSED IN THE SYSTEM

The following is an account of the data processed during e-voting.

The system obviously 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 supervisors, etc.

LOCATION in the following table means that the information is accessible to the particular component at a certain time. B is voter’s browser; audit – auditing system; paper – information published or available on paper. Internet is not a component because information moves in encrypted form and cannot be read by the communication channel.

<table>
<thead>
<tr>
<th>Information generated in the voting process</th>
<th>B</th>
<th>VFS</th>
<th>VSS</th>
<th>VCA</th>
<th>audit</th>
<th>paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polling list</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Candidate list and public additional info</td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voter application</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal information on voter/digital signer</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIN-code of voter’s ID card</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

1) VCA secret key is only open to use, it cannot be copied through VCA server.
9. ANNEX 2 – DATA CHANNELS INTO AND OUT OF THE SYSTEM

Input information

<table>
<thead>
<tr>
<th>Information Type</th>
<th>Source(s)</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate list</td>
<td>NEC</td>
<td>VSS</td>
</tr>
<tr>
<td>Polling list</td>
<td>Andmevara</td>
<td>VSS</td>
</tr>
<tr>
<td>renewals of polling list</td>
<td>Andmevara</td>
<td>VSS</td>
</tr>
<tr>
<td>nullifications and restorations vote</td>
<td>NEC</td>
<td>VSS</td>
</tr>
<tr>
<td></td>
<td>voter</td>
<td>VA</td>
</tr>
</tbody>
</table>

Output information

<table>
<thead>
<tr>
<th>Information Type</th>
<th>Source(s)</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>feedback</td>
<td>VA</td>
<td>voter</td>
</tr>
<tr>
<td>voting status of a voter</td>
<td>VA</td>
<td>voter</td>
</tr>
<tr>
<td>lists of e-voters</td>
<td>VSS</td>
<td>polling divisions</td>
</tr>
<tr>
<td>e-voting results</td>
<td>VCA</td>
<td>NEC</td>
</tr>
<tr>
<td>monitoring results</td>
<td>supervisory application</td>
<td>NEC, system supervisors</td>
</tr>
<tr>
<td>logs, audit results</td>
<td>auditing system</td>
<td>NEC, archive</td>
</tr>
<tr>
<td>archived digital signatures e-votes</td>
<td>VSS</td>
<td>archive</td>
</tr>
<tr>
<td></td>
<td>VSS, VCA</td>
<td>to be destroyed</td>
</tr>
</tbody>
</table>

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
- voter application, 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 on the system
- source code of system components
10. 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 work efficiency/operation speed of each component, application or data carrier. The objective of the following risk analysis is not to itemise all possible variations – a simple cross table of system data and components should suffice for that, while the outcome would be of insignificant practical value – but drawing attention to the most important points.

10.1. Risk classification

There are many possibilities for classifying and presenting the risks: according to the attacker (roll-centered angle); the affected part of the system (architectural angle); the cause (error, attack, environmental impact); the data in risk of exposure (data angle); the chronological order (process angle); the 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 privacy** jeopardise the confidentiality of voting.
- **Risks to the operability** jeopardise the operability and usability of the system.
- **Risks to the reliability** jeopardise the correct proceeding of the e-voting process.
- *In addition we point out the risks of the key management*, which compose a logical entity.

All risk assessments are qualitative.

10.2. Risks to the integrity

10.2.1. Discriminatory errors

*Discriminatory errors* are errors whereby the e-voting system treats some voters differently from 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 (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 logs incorrectly 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 ó and rejects such votes.
- The voter application does not function in Russian-language version of Windows, leaving non-Estonians unable to vote.
- The voter application does not work in the Mac-type computers of the Art Institute, leaving art students unable to vote.

The risk of discrimination 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.
10.2.2. Risks related to Internet use

The normal data exchange between the voter browser/application and the central system cannot be attacked via 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. by typing an address like ww.wvvk.ee),
- technical reasons (DNS errors/attacks, wrong configuration of the voter’s computer, etc.).

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 took place in Estonia (and failed only thanks to the inadequate 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 a fake application to the voter, detect his or her choices, etc.

There is no single never-failing 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 issued ID-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.

**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 recently published security failures in the Microsoft software (in operating systems as well as the browser). Since the voter ID-card is also accessible in the computer at the time of voting, the e-voting moment might be the most attractive.

Internet can also violate the uniformity of the voting process through discrimination or selective obstruction of voters. Thus the network connection could be down in Ida-Virumaa on the e-voting days.

10.2.3. 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 in the polling station and must be checked before the start of the elections.

These data can be changed without authorisation – either by breaking into the server or by the technical administrator of the server. Since this changes the most important input data of the voting process, the result of the voting process, i.e. the vote, will also unavoidably change.

**Exposure of 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 could be falsified, vote and voter privacy violated, some candidates excluded from voting, etc.

**Unauthorised change of candidate list**

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

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

**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 party) will cause problems but does probably 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 choosing process. The most serious problem is probably caused by candidates remaining „off-screen”. The Estonian alphabet might also cause display problems in 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 of variable size, it is extremely easy to create the so-called Florida Butterfly Effect whereby the candidates and the ballot boxes do not line up and confused voters vote for an unintended candidate.

**Classical web applications and web server errors**

These are last on the list because they are well-known and typical, not because they are harmless. 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 repeatedly. A relevant list can 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, alas, a rare luxury in our high-speed 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 which are much harder to detect.

10.2.4. Voter’s computer, web browser, voter application

**Exposure of candidate list**

If the candidate list displayed to the voter is not correct, neither can be his or her vote. 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 key management.

Functional failures of the voter application

An application may contain design errors, inadvertent errors as well as intended Troy horse type 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“ by voter application 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 increases the risk of coming into contact with misleading publicity materials. 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 number 666 on the candidate list is T.T.

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

Intentional sending of an incorrect vote by the voter

The voter application 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 a certificate which does not belong to the particular ID-card; 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 overflows 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 annul the certificate after the vote is sent.

General risks of the signature application

The voter application has all the classical risks of signature application, deriving from the fact that it has access to the voter 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".
10.2.5. 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.

10.2.6. VSS
VSS as the most complex component in the functional sense has the best control over the votes, and thus also the best opportunities to manipulate 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 him or her on that particular day. If someone is missing from or added to the polling list, then either a legal voter is not able to 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 signature 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 the 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 ID-card 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.

10.2.7. 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.
10.2.8. Validity confirmation or time stamping service
Validity confirmation service cannot influence the correctness of the voting in any other way besides discrimination.

Time stamping service has no way of influencing the correctness of the voting. Discrimination is out of the question because TSP only sees the hash of the digital signature with no possibility of deriving any information on the vote or the voter.

10.2.9. Auditing system and auditing application
This subsystem with a checking function cannot influence the integrity of the system.

10.3. Risks to the privacy

10.3.1. Web server/VFS
Violation 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 the name of the voter. If division code request is logged on the basis of a personal identification code, this data is also somewhere. Similar information is contained 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 voter application when choosing a candidate.

10.3.2. Voter’s computer, web browser, voter application
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 above described security problems of the voter’s computer, a voting trace (evidence of connection to the web server) will also remain in the user web browser log.

10.3.3. 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.
10.3.4. VSS, Intranet

Practically all data in the e-voting system – with the exception of the actual votes, being accessible for the VSS only in the encrypted form – could 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 in 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.

10.3.5. 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 and the value of the vote cast. The hash of the vote would enable to find the voter from VCA.

For this, the attacker could monitor the memory used by VCA or take advantage of the errors in the VCA itself.

It is (theoretically) possible for the VCA private key to leak through VCA.

However, we will look at the key related problems under key management.

10.3.6. 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.

10.3.7. System output

Limited confidentiality of e-voting results

Every division can use this formula for every candidate
\[
\text{votes cast at e-voting} = \text{final result} - \text{votes cast by standard procedure}
\]

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 only a few people have voted.

10.3.8. Auditing system and auditing application

Logs are the traditional data leakage sources. 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.

10.4. Risks to the operability

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.

10.4.1. 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.

10.4.2. 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 has always been responsible for his or her arrival to the polling station.

There might be a problem with the huge volume of the voter application which stops it from being downloaded by voters with a slow Internet connection or makes the whole voting process too long.

10.4.3. 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 traditional neutralising methods exist (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 under 4.1.5, Verification of validity of digital signatures.

Denial of service attacks

It is not too unreasonable to claim that until e-voting does not constitute an important part of the voting process, malicious DoS attacks are not a serious problem for e-voting.

This opinion is based on the great incompatibility between the risk taken by the attacker and the motivation behind it. We do not believe that any 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 it is much easier to block an external attack.

The of e-voting time-frame of three 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 Central System.

10.4.4. 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 in case of a failure. 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 weakest link 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).

10.4.5. 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.

10.5. Risks of key management

10.5.1. 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 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 is exposed**, the confidentiality of all votes cast has been violated.

The relevant dangers have also been described in the conception.

The key could be exposed through errors in executing key management procedures, conspiracies 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.

10.5.2. 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 scheme 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.

Also possible is a 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. 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 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 voter application.

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 detecting 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).

10.6. Risks to the reliability

E-voting differs from other 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 generate material for these.

In other words – 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

One can claim that e-voting does not fulfil certain requirements set to public voting – thus, it does not ensure the confidentiality of the vote, correct result or uniformity of the voting.

Complaint – uncheckability 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 public components of the system

The public system components can be attacked with a lot of ado and high visibility. This involves DoS attacks, defacement of web server, modification of voter application, 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.
Imitative attack, deception

The general public can also be mislead by a claim that an attack has been carried out, and falsified proof of it can be presented. 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 the actual appearance of the page, or claim that the voter application 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 have a newspaper article published 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.
11. 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 conceptional improvements recommended in the present analysis are applied.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Location</th>
<th>Probability</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fundamental and process management problems</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks deriving from formalisation of processes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks deriving from centralisation of processes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality of system project – design errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality of system development – software errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System management quality – configuration and management errors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software problems arising from cryptography</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks of voter’s computer as an uncontrollable environment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks connected to AIP use</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible conflicts of conventional and e-voting processes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attacks against key managers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limited confidentiality of e-voting results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reliability risks</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complaint – inappropriateness of the system to public voting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complaint – uncontrollability of the system</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attack against the public components of the system, e.g.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>defacement of web server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imitative attack and deception</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Errors of the auditing system and auditing application</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Risks influencing the correctness of voting results</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directing the voter to a falsified web page</td>
<td>net</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>Man-in-the-middle attacks between the web server and the voter</td>
<td>net</td>
<td>x</td>
<td>xx</td>
</tr>
<tr>
<td>application</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attacking the user’s computer and gaining control over it</td>
<td>net</td>
<td>xx</td>
<td>xx</td>
</tr>
<tr>
<td>Functional failures of the voter application</td>
<td>PL</td>
<td>xxx</td>
<td>x</td>
</tr>
<tr>
<td>Use of misleading data by the voter</td>
<td>PL</td>
<td>xx</td>
<td>x</td>
</tr>
<tr>
<td>Intentional sendoff of an incorrect vote by the voter</td>
<td>PL</td>
<td>xx</td>
<td>x</td>
</tr>
<tr>
<td>Votes signed with an invalid certificate taken into account in the</td>
<td>PL</td>
<td>x</td>
<td>xx</td>
</tr>
<tr>
<td>results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General risks of signature application</td>
<td>PL</td>
<td>xx</td>
<td>x</td>
</tr>
<tr>
<td>Unauthorised changing of input and output data in the system</td>
<td>Central system</td>
<td>x</td>
<td>xxx</td>
</tr>
<tr>
<td>Jeopardising voter application</td>
<td>VHS, PL</td>
<td>x</td>
<td>xxx</td>
</tr>
<tr>
<td>Substitution of the VCA public key in voter application</td>
<td>VHS, PL, key</td>
<td>x</td>
<td>xxx</td>
</tr>
<tr>
<td>management</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## E-voting conception security

<table>
<thead>
<tr>
<th>Potential Security Threats</th>
<th>Source(s)</th>
<th>Impact</th>
<th>Risk</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faulty candidate list/ unauthorised changing of candidate list</td>
<td>VHS, PL</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Static content of the web server jeopardised</td>
<td>VFS</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Typical web applications and web server errors</td>
<td>VFS</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Errors of input data in VSS</td>
<td>Database</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Unauthorised changing of input data in VSS</td>
<td>Database</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unauthorised changing of other data in the database</td>
<td>Database</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing the list of unsigned votes</td>
<td>Intranet</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Functional errors of VSS applications</td>
<td>VSS</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Digital signatures checking errors</td>
<td>VSS</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Functional failures of the voter application</td>
<td>VCA</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Discriminatory errors</td>
<td>All components</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

### Risks influencing the confidentiality of votes or voting result

<table>
<thead>
<tr>
<th>Potential Threat</th>
<th>Source(s)</th>
<th>Impact</th>
<th>Risk</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Violation of the fact of voting in voter’s computer</td>
<td>PL</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Violation of confidentiality of vote in voter’s computer</td>
<td>PL</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Violation of the fact of voting in the Internet</td>
<td>Net</td>
<td>xxx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Violation of the fact of voting in the Central System</td>
<td>VFS</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Violation of confidentiality of vote in web server</td>
<td>VFS</td>
<td>xx</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Violation of confidentiality of vote in VSS</td>
<td>VSS</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Leakage of the complete data base of votes</td>
<td>VFS, VSS, Intranet</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Leakage of votes from vote counting application</td>
<td>VCA</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Leakage of the list of e-voters through the validity confirmation service provider</td>
<td>VCS</td>
<td>(xx)</td>
<td>(x)</td>
<td></td>
</tr>
<tr>
<td>Leakage of information on e-voting usage intensity</td>
<td>Net</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Exposure of VCA secret key</td>
<td>VCA, key management</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Leakage of logs from the auditing system</td>
<td>Audit</td>
<td>xx</td>
<td>xx</td>
<td></td>
</tr>
</tbody>
</table>

### Factors influencing the efficiency of the voting system

<table>
<thead>
<tr>
<th>Factor</th>
<th>Source(s)</th>
<th>Impact</th>
<th>Risk</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors in system management</td>
<td>Central system</td>
<td>xx</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Servers and communications network operability risks</td>
<td>Central system</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Failures and quality problems of central system software</td>
<td>Central system</td>
<td>xx</td>
<td>xxx</td>
<td></td>
</tr>
<tr>
<td>Failures of central system hardware</td>
<td>Central system</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Incorrect planning of necessary system resources</td>
<td>Central system</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Failures and quality problems of voter application</td>
<td>VA</td>
<td>xxx</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Vast volume of the voter application (regarding the network connection of voters)</td>
<td>VA</td>
<td>xx</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Failures of the data base operability</td>
<td>VSS</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Non-operability of validity confirmation/time stamping service</td>
<td>VCS</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Service constraint attacks</td>
<td>Net, VFS</td>
<td>x</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Central system overload caused by voter/ VA</td>
<td>VFS, VSS</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>VCA application failures</td>
<td>VCA</td>
<td>xx</td>
<td>xx</td>
<td></td>
</tr>
<tr>
<td>Destruction/ inaccessibility of VCA secret key</td>
<td>Key management</td>
<td>x</td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>Operability of VCA pirate key (decrypting speed)</td>
<td>VCA</td>
<td>xx</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
12. ANNEX 5 – SECURITY MEASURES DEEMED UNNECESSARY

The working group discussed the following security measures but did not set these as requirements. Their implementation is, of course, not prohibited.

**VA – VFS communication additional security features**

An additional security layer could be added to VA-VFS communications:
- voter application could check whether the web page holds 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 forging 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 a 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, thus basically a paper copy of Logl.

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 artificial meaning.

**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 attainable added security goes nowhere near of compensating for them.

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

Proceeding from the system architecture, there can be an unlimited amount of central system components, excluding 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-span 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.

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 ends up reducing the operability 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 referred to in the conception.

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 1\textsuperscript{st} or the 2\textsuperscript{nd} key.
- **We do not know if the 2\textsuperscript{nd} 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 system detecting this.

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

**Closed VA programming – protocol is public, application is not**

Voter application can be written in a way which makes its deconstruction more complicated, and its code can be published only at the last moment before the start of the voting. This complicates considerably the attacks aimed at compromising the application – disassembly of the application, creation and distribution of a new one will take so long that there simply will not be enough time to carry out the attack.

Firstly, this is politically incorrect in case of a system with an open design; secondly, applications of previous votes still exist; thirdly, protocols are public and the attacker can simply write their own application for the attack.

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

Alleviates various abuse risks but creates even more. There is also no reasonably suitable channel.

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

We discussed many ways for authenticating the users of the central system and zoning the network. The users could thus be engaged in a separate network and allowed access into servers only through a firewall. This would ensure 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 sound file**

This would have avoided 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 only a small number of e-votes cast (cf. chapter 11.3.7, System output) the value of a vote could be exposed through exclusion method. This can be avoided by demanding that in case of low voting activity e-voting would 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 accept the result.

We did however reach the conclusion that this was unnecessary.
13. ANNEX 6 – REFERENCE WORKS

[Florida] The Butterfly Ballot: Anatomy of a Disaster
Ask Tog webzine, January 2001

[IP1-S-EE] “Recommendation on legal and operational standards for e-enabled voting (Second Draft)”
IP1-S-EE working group, July 2003
www.coe.int

[ISKE] Infosüsteemide kolmeastmelise etalonturbe süsteem ISKE,
Cybernetica AS, October 2003

[Neumann] Security Criteria for Electronic Voting
Peter G. Neumann, September 1993
http://www.csl.sri.com/users/neumann/ncs93.html

[OWASP] The Open Web Application Security Project Top Ten Vulnerability List
http://www.owasp.org/documentation/topten

[PKCS] PKCS #1 v2.1 RSA Cryptography Standard
RSA Laboratories, June 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, July 2003
http://avirubin.com/e-voting.security.html