Threat Assessment Summary

e-Voting, *Admin*, and *pVoting* TOE's

DOCUMENT HISTORY

Version	Date	Author	Comments
0.1	2011-04-06	Scytl R&D	Created Document.
0.9	2011-04-20	Scytl R&D	First full version for review
1.0	2011-04-27	Scytl R&D	First version
1.1	2011-06-02	Scytl R&D	Added a rationale of low risks
1.1	2011-06-05	Scytl R&D	Some risk descriptions have been modified.

DISCLAIMER: This document is a work in progress and some information in it might be obsolete, inaccurate or might be missing. Regular updates will be made to the document. This disclaimer will be also updated to reflect the state of the document.

QUALITY ASSURANCE:

Version	Date	QA responsible	Comments
1.0	2011-04-27	Project Manager	Approved
1.1	2011-06-06	Project Manager	Approved

PATENTS, RELEVANT TO THE SOFTWARE, ARE LICENSED BY SCYTL SECURE ELECTRONIC VOTING SA TO THE NORWEGIAN MINISTRY OF LOCAL GOVERNMENT AND REGIONAL DEVELOPMENT FOR THE PURPOSES SET OUT ABOVE.

SCYTL SECURE ELECTRONIC VOTING SA (OR WHOM IT APPOINTS) HAS THE RIGHT, INSIDE AND OUTSIDE OF NORWAY TO USE, COPY, MODIFY AND ENHANCE THE MATERIALS, AS WELL AS A RIGHT OF LICENSING AND TRANSFER, INTERNALLY AND EXTERNALLY, EITHER BY ITSELF OR WITH THE ASSISTANCE OF A THIRD PARTY, AS PART OF THE FURTHER DEVELOPMENT AND CUSTOMIZATION OF ITS OWN STANDARD SOLUTIONS OR DELIVERED TOGETHER WITH ITS OWN STANDARD SOLUTIONS.

THE NORWEGIAN MINISTRY OF LOCAL GOVERNMENT AND REGIONAL DEVELOPMENT AND SCYTL SECURE ELECTRONIC VOTING SA HEREBY GRANT TO YOU (ANY THIRD PARTY) THE RIGHT TO COPY, MODIFY, INSPECT, COMPILE, DEBUG AND RUN THE SOFTWARE FOR THE SOLE PURPOSE OF TESTING, REVIEWING OR EVALUATING THE CODE OR THE SYSTEM SOLELY FOR NON-COMMERCIAL PURPOSES. ANY OTHER USE OF THE SOURCE CODE (OR PARTS OF IT) FOR ANY OTHER PURPOSE (INCLUDING BUT NOT LIMITED TO ANY COMMERCIAL PURPOSES) BY ANY THIRD PARTY IS SUBJECT TO SCYTL SECURE ELECTRONIC VOTING SA'S PRIOR WRITTEN APPROVAL.

1 Threat Assessment Methodology

A threat assessment methodology has been established in order to identify and prioritize the security issues – both attacks performed by internal or external attackers, and incidents regarding software or hardware components - which could be damaging the Voting Platform.

This study has been performed from the "Security Target" documents on the Common Criteria basis. Reading this "Security Targets" is necessary to understand this threat assessment summary, since you need to know at least the solution architecture.

1.1 Secure Usage Assumptions

To delimit the possible software security threats to the technological ones, the following secure usage assumptions have been considered:

Assumptions	Description
A. Authentication	It is assumed that a secure voter identification and authentication service will be provided by the client (MinID).
A. Physical	It is assumed that TOE servers are installed in a physically secure location that can only be accessed by authorized users.
A. Secure Deployment and Operation	It is assumed that the TOE and its dependencies (E.g. operating system) are installed and managed in a secure way.
A. Timestamp	It is assumed that the TOE environment uses reliable synchronized time sources.
A. Trusted Administrator	It is assumed that the administrators are well trained and follow all administrator guidelines.
A. MonitoringTask	Although the TOE generates logs for every critical action and security issues of the TOE, it is assumed that operators shall perform monitoring tasks, and they will be responsible for responding to security incidences and reporting them.
A. Contingency Plan	It is assumed that a documented plan is provided to maintain continuity of operation in an emergency or disaster.
A. Controlled Environment	It is assumed that the Voting Terminals at the controlled environment polling places have been properly hardened, so they are free of any malicious software, and they are always running the authorized version of the voting software (voting applet).

1.2 Threat Assessment methodology

The following steps have been followed during the threat assessment process:

1. **Threats identification**. The threats have been identified during the solution design using: the sequence diagram of processes, software module, user history, use case…
2. **Threats analysis**. Detailed specification of security threats, including the analysis of the attackers who can perform the action, the assets damaged by the threat, and the impact produced over the solution. This specification includes the following fields:
    - Phase. Election phase where the threat has been identified.
    - Agent. Agent that exploit the threat (external attacker, system administrator, voter, …).
    - Affected Assets. Assets compromised by the threat (vote, ballot box, database…).
    - Threat. Specific threat description.
    - Complexity. The effort complexity required to perform the attack (numeric value).
    - Impact Description. Description of the consequences related to the threat.
    - Impact. Quantification of the consequences related to the threat (numeric value).
3. **Security Controls identification and assessment.** The countermeasures that are preventing or mitigating the security threats. Fields used:
    - Control Responsible. The software module, infrastructure, operating system, manual procedure... responsible of implementing the countermeasure.
    - Control Type. It could be a preventive, detective, or corrective control.
    - Control. Countermeasure description (in a comprehensive language).
    - Control Effectiveness. Percentage of control effectiveness - from 0% to 100%.
4. **Risk Calculation.** The numerical analysis of existing risks using the numeric values from previous steps:
    - Initial risk (or Intrinsic Risk) is the calculated Risk of each threat without applying any control (Complexity * Impact).
    - Final risk (or Residual Risk) is the calculated Risk of each threat applying selected controls (Complexity * Impact * Control effectiveness). As the 100% security cannot be assumed, a 95% rate is applied.

1.3 Risk levels

As said before, the risk values have been calculated from the numeric values assigned to their "Complexity" and "Probability". However, to "translate" these numbers to better understand consequences over the elections, the following risk levels have been used:

- **VERY CRITICAL RISK**. There are no controls to mitigate the threat and it is very probable that it happens, having a terrible consequence over the election.
- **CRITICAL RISK**. There are no controls to mitigate the threat and it is less probable that it happens, having an adverse consequence over the election.
- **MEDIUM RISK**. Security controls for detecting and/or preventing the risk are in place, but the risk could happen in some cases if one of these controls fails.
- **LOW RISK**. The threat is controlled by redundant countermeasures that prevent a single control failure to expose the system, but there are a few combinations of control failures that could be used to exploit the system.
- **VERY LOW**. The situation is under control and only a large amount of control failures or a complete disaster could expose the system to the risk. Since this very improbable to happen, the risk is considered insignificant.

It is important to consider that is really impossible to have something 100% secure. For this reason, when calculating the final risk a reduction coefficient have been applied to ensure the controls were not mitigating 100% the risks - but up to 95% maximum. Therefore, there is not any threat with risk value equal to 0.

2 Threat Assessment summary of results

2.1 Top-Risk summary:

From the detailed analysis of the 3 TOEs, **170 different security attacks** were identified at some components of the TOEs. All these attacks have been **evaluated individually**, and – after considering the security controls in place – we can conclude that:

- **None of them has been considered as "very critical" or "critical"**.
- Only 7 of them (4%) have been categorized as "medium risk", where medium risks are representing possible situations which are already mitigated by existing security controls.
- 12 of them (7%) have "low risk".
- The risk of the remaining attacks is very low or not significant.

The following table represents the security risks by TOE:

SECURITY RISKS					
TOE	VERY CRITICAL	CRITICAL	MEDIUM	LOW	VERY LOW
e-Voting	0	0	5	8	99
Admin	0	0	2	4	40
pVoting	0	0	0	0	12
TOTAL	0	0	7	12	151

Considering this analysis, we can conclude that regarding the three TOEs the security level of the applications platforms is VERY HIGH.

The higher risk situations are related to:

- Authentication on the controlled environments.
- The audit of the mixing process.
- Malware on the voters' PCs.
- Intercepting the electoral roll information between the admin server and the authentication server.
- Direct access to the Admin system databases.

2.2 Top-Risk List:

The following are the identified *Medium* and *Low* risks (there are not any critical o very critical risks).

TOE	Risk	Level (*)
e-Voting	Authentication Token from the smart card can be forged.	**MEDIUM**
e-Voting	A malware accesses to the selected voting options at the Voter PC	**MEDIUM**
e-Voting	Authentication Service - Voter Impersonation and Vote Casting	**MEDIUM**
e-Voting	Man-in-the-Middle Voter Contest Modification between the electoral roll service and the authentication service	**MEDIUM**
e-Voting	A malware modifies the ballot template at the Voter PC	**MEDIUM**
Admin	Direct system access to back end Admin server or Admin database	**MEDIUM**
Admin	Direct system access to electoral roll database	**MEDIUM**
Admin	Leave Admin client PC with active session	**LOW**
e-Voting	Authentication Server redirection to a fake Electoral Roll	**LOW**
e-Voting	Man-in-the-middle- Ballot template modification between voter and voting servers	**LOW**
e-Voting	A malware modifies the voting options at the Voter PC	**LOW**
e-Voting	Mixing Process is changing Votes	**LOW**
e-Voting	DoS attack over the Voting Platform (Denial of Service)	**LOW**
Admin	DoS attack against the electoral roll (Denial of Service)	**LOW**
e-Voting	Erroneous mixing process (shuffling is not properly performed)	**LOW**
e-Voting	Decrypted Ballot Box Modification at Counting	**LOW**
Admin	Audit data loss, or logs are lost due to a system failure (audit systems not available)	**LOW**
Admin	Man-in-the-Middle attack on Admin web services	**LOW**
e-Voting	Controlled environment Authentication Token reply attack	**LOW**

(*) The numeric values of the risks are not shown for making it easy to understand, but risks have been sorted in this table according to these values.

2.3 Top-Risk Rationale:

In this chapter, the main detected risks (medium and low category risks) are explained.

2.3.1 Medium Risks

- **At the Controlled Environment, the Authentication Token from the smart card can be forged.**

When voting at the controlled environment, the voter presents her identity card to the electoral officer, who creates an authentication token - linked to the voter SSN - digitally signed by an application key installed in the poll-site computer. This token is recorded into a smartcard that is introduced at the voting terminal to authenticate the voter.

The following facts could expose the voter authentication process at risk in the controlled environment:

- The digital certificate (in an encrypted P12 file format) from the poll-site computer could be exported to another computer.
- The password required to decrypt the private key from the P12 file, is known by the electoral officials.
- The authentication token is recorded at the public zone of the smartcard, so the PIN of the smartcard is not required to record at the authentication token.

Considering all these facts, a malicious electoral official could try to export the P12 file to another computer which could be continuously registering fake authentication tokens at forged smartcards. If these fake smartcards are used in the voting terminal the votes cast with them would be considered valid votes.

Moreover, no technological controls would be detecting this attack, since the voter is not receiving the return codes by SMS but in the screen.

However, this situation is mitigated by procedural controls in the polling-places: if voters authenticated by their national id-card are only granted to cast one vote, any anomalous voter or election official behavior would be detected (someone trying to cast votes with several smartcards, or someone trying to access to the voting terminal without authentication). Furthermore, as mentioned before, this tokens cannot be used to cast a vote from an uncontrolled environment (i.e., from personal computers).

- **Malware access to the selected voting options at the Voter PC**

Since the vote is being cast from a PC, and any PC is susceptible to have malicious software (malware) installed without the knowledge of the voter, any activity performed by the voter from her PC (e-voting or any other personal operation) could be "observed" by a malware.

The only mitigation control for this risk, is the "Multiple voting" feature. In this case, a malware "observing" the voter activities from her PC would not be sure about the observed vote to be the counted vote, since the voter is able to cast another vote from a different computer or at a poll-site (electronically or by paper).

- **Authentication Service - Voter Impersonation and Vote Casting**

The following elements are requested to cast a valid vote:

- An identification token (from the IDPorten service or a poll site officer).
- An authentication token created by the Authentication Service.
- A vote digitally signed by the voter – with the voter credentials from the P12 file.
- A voting receipt generated and signed by the RCG.

In the hypothetical case that the Authentication Service was compromised and was behaving maliciously (manipulated software or an attacker operating from the server), it could try to simulate an error to force a voter to identify two times in the IDPorten. Therefore, it could cast a valid vote by keeping an unused identification token (the first one), constructing a valid authentication token digitally signed by itself, and sending a valid vote digitally signed with the voter private key (since the Authentication Service is storing the P12 files and the passwords of the P12 are trivial passwords) after the valid vote of the voter.

This attack could be feasible only for remote votes – not for votes cast from the poll sites, since in the last case only one vote is granted per voter, and an identification error would be investigated.

In case the attack happens, neither the VCS nor the RCG or the cleansing service could detect it when verifying these votes.

However, the residual risk is being reduced since voter will be always aware of it since she will receive an unexpected SMS message with the return codes which are sent to the Voters. In addition, the software installed in the servers is logically sealed and baselined before the election starts to detect any manipulation (e.g., intrusion that could change). The status of the sealing is verified periodically and compared with the initial baseline.

- **Man-in-the-Middle - Voter Contest Modification between the electoral roll service and the authentication service.**

Each time a voter is registered at the authentication service, it queries to the electoral roll service to obtain the contests the voter is eligible to cast a vote.

However, when the electoral roll service replies the query with the contests information, it is not currently possible to ensure the authenticity of the message since:

  - The connection between the electoral roll service and the authentication service is not being authenticated by SSL.
  - The electoral roll service does not digitally sign the contests information before sending it.

Therefore, in case an attacker could impersonate the electoral roll service, she could try to make voters to cast a vote for a contest that they are not officially eligible for, or she could try to prevent eligible voters from casting a vote for a contest they are eligible for. If the attack happens, and it is not noticed by the voter or the voter collaborates to implement it, the authentication server would not be able to detect these situations.

This attack is mitigated by the following security controls:

- Both the authentication service and the electoral roll service are located in the same LAN, so this attack is only feasible by inside attackers.
- The contests authorizations will be also verified at the Cleansing Service, so finally the fraudulent votes would not be counted even though there were accepted in the casting process.

Therefore, this attack cannot generate an invalid vote that will be counted but could prevent a vote from being counted if the voter did not noticed that he was not casting a vote for her eligible contest.

- **A malware modifies the ballot template at the Voter PC**
- **A malware modifies the voting options at the Voter PC**

Since the vote in uncontrolled environments is being cast from a voter PC, this computer is susceptible from having malicious software (malware) installed without the knowledge of the voter. Therefore, any activity performed by the voter from her PC (e-voting or any other personal operation) could be "manipulated" by a malware.

In this case, when receiving the ballot template (the XML file with the voting options from the Vote Collector Server) at the voter PC, this file is susceptible from being modified by a malware inside the voter PC, affecting to the content of the ballot template. By this way, a malware inside the voter PC could shift parties and candidates' positions to cheat the voter.

The voter can detect this situation when verifying the return codes from her voting options. However, since these return codes are only affecting to parties and not to candidates (but candidate position) inside each party, this attack will be detected when the voter selections are related to parties or responses to questions, since these are not based on position codes but party codes. In case the return codes are related to candidate positions, voter will receive the correct position codes of the selected voting options as shown in the selection screen. However, if the voter PC has been compromised, there is still a remaining risk that the position of the candidates differs from their positions in the official ballot and therefore, the final selection does not really represent the voter intent in the same party list.

The unique control to mitigate this attack is procedural: voter can detect this situation when verifying the return codes from her voting options against an official ballot (the position codes will reveal that candidates in the selected positions are not the same as intended). That way, the voter is able to use another voting terminal (or voting from a polling station) if the return codes shown this issue.

Summarizing, the attack will be effective only in case the following two conditions are satisfied:

- The malware modifies the position of candidates inside the same party in the ballot template before being displayed to the voter but does not modify the parties.
- If the voter selects at least one individual candidate inside the selected party or selects at least one candidate from another party.

In the other hand, the platform can be configured to return to the voter only selected party codes and verbally the number of selected candidates per party instead their position return codes. When the platform is configured that way, the malware does not need to shift candidates on the ballot template, but only to change the selected candidates in the vote to be encrypted by other candidates of the same party. If so,

changes will not be detected by checking any official ballot since there are not return codes that state the selected positions (voters are only noticed about the number of selected candidates per party). As in the previous case, this attack can only be implemented if the voter selects candidates in addition to the party. Therefore, the risk of using this configuration needs to be carefully evaluated in the context of the election (e.g., since this attack does not change the number of votes given to a party, the attack has no impact when this is the value used to obtain the results).

In any case, the impact of this residual risk would be limited to candidates' changes inside the same party – no party changes. Furthermore, this attack is very difficult to implement compared with the final impact, since requires to develop a specific malware that needs to be widespread on Internet, successfully infect computers without being detected, and shall be able to discern if the computer belongs to a voter and when it is used to cast a vote (and make modifications on the fly).

- **Direct system access to back end Admin server or Admin database**.
- **Direct system access to electoral roll database.**

In case an internal attacker accesses to the Admin database, she could try to modify critical information required for different components of the platform, like the candidates lists, the electoral roll, the poll-sites configuration, the paper votes or e-votes counts, …

Since the information is not stored digitally signed at the Admin Database, any direct modification to the database would be only detected by manual reviews (which are sometimes complicated due to the large amount of data).

By modifying without proper authorization the information contained at the Admin database, the election configuration information could be affected causing the election results to be invalidated, even the election results could be altered by adding or removing voters from the electoral roll.

The mitigation of this risk is mainly based on restricting and controlling the access to the servers.

2.3.2 Low risks

- **Leave Admin client PC with active session**

The PCs where the "Admin application" is managed by electoral officials are susceptible to be used by an internal attacker (insider).

The access to these PCs is restricted by authentication through username and a password; when the electoral officials leave their work place unattended, even if they do not lock manually their computers, these are automatically locked after a certain inactivity period (configurable). However, before this configurable period of time the computer is susceptible of being used by an insider, gaining access to the "Admin application" – if the user is still logged-in – to modify critical election configuration data.

Although the residual risk exposed by this threat is low, the administrative people managing the "Admin" application shall be aware of such risk and be instructed about locking manually their PCs when leaving their workstations.

Additional unknown controls could be helping to mitigate this risk, like the location of the computers in open-spaces where an insider using a computer without being the owner would be detected by other work-colleagues. Although these controls have not been considered in the risk assessment, they can be considered in this rationale.

- **Authentication Service redirection to a fake Electoral Roll**
- **Man-in-the-Middle attack on Admin web services**

The Authentication Server is configured to access the Admin Server, where the application "Electoral Roll" is located. By this way, when a voter is identified – by IDPorten or a poll site officer – the Authentication Service verifies the contests she is granted to vote according to the electoral roll definition.

If someone achieves to impersonate the Electoral Roll Service inside the internal network of the datacenter, by changing the configuration from the Authentication Server, (e.g., attacking the network, or performing a man in the middle attack) the Authentication Service could obtain fraudulent information, granting the right to vote for an unauthorized voter or denying this right to an eligible one.

Since the connection between the Authentication Service and the Electoral Roll Service is not protected by SSL, and the information transmitted by the Electoral Roll Service is not digitally signed, the Authentication Service is not able to detect the impersonation.

Even in the case that the voter was not registered at all in the Electoral Roll, which means that she has not a valid credential assigned, if the Authentication Service receives the

instruction from the Electoral Roll application to authenticate the voter, the voter would receive a spare credential allowing her to cast a vote.

However, the following security controls are protecting the platform to detect this attack:

- The attack should only be performed from the internal network, which is restricted to internal personnel.
- The access to the Authentication Service for changing the redirection to the Electoral Roll Service is restricted and changes are monitored.
- Once the e-voting period has finished, the cleansing service verifies all the votes to ensure the voter is granted to cast the vote, using the valid Electoral Roll data digitally signed by the Admin system.

Therefore, even though the attack could be possible, it will not remain undetected and the invalid votes will not be counted.

- **Man-in-the-middle- Ballot template modification between voter and voting servers**

When the voter is authenticated in the e-voting application, the voting applet running in the voter PC receives a ballot template with the voting options that the voter is eligible to vote.

If an external attacker – between the voter PC and the voting servers – intercepts and modifies this ballot template (e.g., changing the order of the parties or candidates) the voter would be casting a vote with a different content than the intended one.

Although this modification of the ballot template would not be detected by the voting applet (this information is not digitally signed), the following security controls have been implemented for preventing and detecting this attack:

- An SSL connection is established between the voter PC and the servers, to ensure the authenticity and integrity of the transmitted information – including the ballot template. A man-in-the-middle attack should break this SSL to be successful, and the voter would notice this in her internet browser.
- The selected parties are confirmed through return codes which can be reviewed by the voter, so she would detect the change.

There is a residual risk regarding this attack since there are not return codes specific per candidate but per candidate position (such as previously introduced in the risk when “A malware modifies the ballot template at the Voter PC”). This attack requires:

- An external attacker breaking the SSL security, and– intercepting and modifying only the order of the candidates inside the parties, but not the parties.
- A voter selecting some individual candidates inside the parties.

Considering both conditions are satisfied, the same attacks described when "A malware modifies the ballot template at the Voter PC" applies in this case.

In any case, the impact of this residual risk would be limited to candidates' changes inside the same party – no party changes.

- **Mixing Process is not behaving properly**

The mixing activities are generating proofs for the validation the correct behavior of the process. However, these proofs are not forced to be validated by the counting process when the Mixing process finishes. Although an auditor is able to verify them subsequently, the counting process assumes that an auditor is verifying these proofs, and proceeds with the counting when it receives the Mixing output. The auditor should verify and approve the mixing proofs in a different operation flow to ensure that the results are correct.

In case the mixing process wrongly implemented the mixing process the output could contain invalid votes or missed valid ones.

The currently implemented mitigating controls are implemented to reduce the risk of Mixing issues:

- The mixing process is performed in an isolated and controlled environment, which is ensuring nobody is able to modify the software or data.
- An auditor is able to verify the proofs generated by the nodes, ensuring the right mixing of the votes. This verification would detect the situation for repeating the mixing and counting processes.
- The proofs cannot be generated without using a random challenge that is provided by the auditor at the end of the Mixing process, when all the input/outputs of the nodes are provided and therefore, cannot be altered.

Therefore, in addition to the generation of audit proofs for detecting any disruption of the Mixing process, additional controls are implemented to make any issue more difficult to implement and reduce the risk.

- **DoS attack over the Voting Platform (Denial of Service)**
- **DoS attack against the electoral roll (Denial of Service)**

A denial of service attack consists on sending so many requests to the targeted servers than they collapse and they were not able to manage valid requests; even the collapse reaches to cause a serious server failure.

Although the platform infrastructure has been estimated to support a big quantity of requests at the same time and is monitoring malicious behaviors (load balancers, firewalls, IDS…), a distributed denial of service attack with thousands of malicious computers sending thousands of requests at the same time could set the e-voting service unavailable during a period of time.

However, since the election process is open several days, the impact of the attack will be mitigated since such attacks cannot be sustained for a large amount of time (1 or 2 days). In the other hand, as a contingency measure voters are able to vote in any polling place near to their location, since vote anywhere is enabled. Since connections to these polling places are independent from the other ones and not public, the DoS attacks will not affect them.

- **Erroneous mixing process (shuffling is not properly performed)**

The mixing activities are generating proofs for the validation the correct behavior of the process. However, these proofs are not forced to be validated by the counting process when the Mixing process finishes. Although an auditor is able to verify them subsequently, the counting process assumes that an auditor is verifying these proofs, and proceeds with the decrypting and counting when it receives the Mixing output. The auditor should verify and approve the mixing proofs in a different operation flow to ensure that the results are correct.

In case all nodes from the mixing process were malicious or were wrongly implemented the mixing process might be able to do not shuffle the votes, keeping the original order to link the voter with their votes and breaking the voter privacy.

The currently implemented mitigating controls are the following:

- The mixing process is performed in an isolated and controlled environment, which is ensuring nobody is able to modify the software or data.
- To perform this attack is necessary not only to modify the mixing software; the attacker shall have access to the original ballot box (from the cleansing, the VCS, or the air-gap media) and also to the decrypted votes.

- An auditor is able to verify the proofs generated by the nodes, ensuring the right mixing of the votes. If this verification is performed before the decryption process, the situation could be detected for repeating the mixing processes.

These controls contribute to make any attack, even though detectable, difficult to implement.

- **Decrypted Ballot Box Modification at Counting**

Once the votes have been decrypted at the counting server, they are stored in a database and digitally signed in a file. Additionally, zero knowledge proofs are generated for the decryption validation.

However, when the counting process is performed, for performance reasons, the votes are read directly from the database instead of the digitally signed file, being processed without verifying if the digital signature of the file still corresponds to the contents of the database. Therefore, there is a risk that the votes of the database are manipulated after the file is digitally signed and before being counted (race condition).

If an internal attacker achieves to modify the decrypted votes, the election results could be potentially altered.

The residual risk is low, since:

- The counting process is performed in an isolated and controlled environment, which is ensuring nobody is able to modify the software or data,
- Any auditor will detect this attack when verifying the election results by checking the decrypting zero knowledge proofs, the digital signature of the decrypted votes, and the counting results. In case the auditor detects some issue, the decryption and counting process can be performed again with the correct votes.

Therefore, these measures in addition to make the attack detectable also make the attack more complex.

- **Audit data loss, or logs are lost due to a system failure (audit systems not available)**

All components from the voting platform – both the application and the infrastructure ones – are generating logs which are stored locally in each component; just when generated they are sent to an intermediate log gateway (one per datacenter) and automatically forwarded to the Central Audit Server where all logs are stored and monitored.

The risk of losing audit data is low since there are several copies of the logs in different components. However, there is some residual risk due to the Central Audit Server is not

deployed in High-Availability, and in case of a security incident in the Central Audit Server it would be required to gather manually the original information from the other servers.

- **Controlled Environment Authentication Token reply attacks.**

When voting at the controlled environment, the voter presents her identity card to the electoral officer, who creates an authentication token - linked to the voter SSN - digitally signed by an application key installed in the poll-site computer, and the token is recorded into a smartcard. Then, this smartcard is entered at the voting terminal to authenticate the voter.

Since the authentication token is recorded at the public zone of the smartcard, where the PIN of the smartcard is not required, someone accessing to the smartcard (e.g. stealing it) would be able to duplicate the information from the smartcard as many times as they want.

However this risk is mitigated by the security control implemented in several components of the e-voting platform (authentication service, VCS, RCG, and cleansing service) which reviews the identification and authentication tokens against an already-used list to ensure the same token is not being used 2 times.

Therefore, the attack cannot generate a manipulated vote that will be counted.