Federal Office
for Information Security

Protection Profile for E-Voting Systems for non-political Elections

BSI-CC-PP-0121

Change history

Version	Date	Description
1.0	2023-12-1	Initial release

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
Phone: +49 22899 9582-0
E-Mail: online-wahlen@bsi.bund.de
Internet: https://www.bsi.bund.de
© Federal Office for Information Security 2023

Table of Contents

List of Figures

List of Tables

1 Protection Profile Introduction

1.1 Protection Profile Reference

Title	Protection Profile for E-Voting Systems for non-political Elections
Short title	BSI-CC-PP-0121
Version	1.0
Date	2023-12-12
Sponsor	Federal Office for Information Security, Germany
Editor	Evaluation Facility of Deutsche Telekom Security GmbH
Registration	Federal Office for Information Security, Germany
Certification ID	BSI-CC-PP-0121
CC Version	CC:2022 Revision 1
Conformance Claim	CC Part 2 extended CC Part 3 conformant

1.2 Protection Profile Overview

The scope of this Protection Profile is to describe the functionality of an e-voting system for non-political elections in terms of [6] and to define functional and assurance requirements for such a system.

Thereby, the Protection Profle utilizes a modular approach that allows the description of the major functionality that can make up an e-voting system by means of Common Criteria.

Therefore, this Protection Profle is structured into the following areas:

- The **base Protection Profile** contains all threats, OSPs, assumptions, objectives and SFRs that concern the e-voting systems in general. The base PP must be used alone for systems that consist of only one single server component.
- The **package for multi-component server architectures** contains additional threats, objectives and SFRs that shall be considered if a TOE consists of more than one server component. The conformance to the base PP shall be unaffected. These interconnected components can be distributed over separate locations.

1.3 TOE overview

An e-voting procedure for non-political elections, for example the election of the Equal Opportunities Officer, includes the processes required for an e-voting during the preparation phase, the execution phase, the evaluation phase and the post-processing phase.

An e-voting implements a set of the following election principles, depending on the electoral regulation:

- Universal: This principle requires that access to the election must always be guaranteed regardless of characteristics such as gender, race, wealth, or religious affiliation. Irrespective of this, there may be restrictions, for example in an association election where only members may vote.
- Direct: This principle means that those to be elected are elected by the voters by means of direct election and in particular not through intermediaries such as Electoral College.

- Free: This principle demands that voters be able to make their own electoral choices without coercion or undue influence.
- Secret: This principle states that the voting process must be conducted in such a way that third parties cannot trace how people voted, besides the final results.
- Equal: According to this principle, each vote cast has the same weight and must have the same influence on the composition of the election result.
- Public: This principle means that essential parts of the election, such as in particular the correct counting, must be verifiable by the public.

The TOE is responsible for only some of the above-mentioned election principles because not all of them can be equally ensured by the TOE. In particular, the principles “universal”, “direct” and “free” are based on organizational requirements, not all of which can be fully controlled by the TOE. Additionally, the TOE may not apply the principle “equal”, according to the electoral regulation.

1.3.1 TOE Type

The target of evaluation (TOE) is a server software consisting of one central server component for conducting secret non-political e-votings, e.g. for the election of the Equal Opportunities Officer, which implements the process of conducting the election.

Accordingly, security functional requirements are defined for the electoral phases, see [5]:

1. Preparation phase (processing of election-related configuration data),
2. Execution phase (registering the electoral acts),
3. Evaluation phase (counting of votes and determination of the election result), and
4. Post-processing phase (export of election execution data).

The requirements explicitly do not refer to organizational election preparation (such as the creation of the voters’ register) and the archiving of election execution data.

1.3.2 Usage and major Security Features of the TOE

The TOE is a product that realizes all its functions on one component. Remote endpoints (not part of the TOE) access the services of the TOE remotely via a secure connection enabled by the TOE (cf. Figure 1).

Figure 1: TOE access model

The TOE implements the election organizer's specifications for the type of display, in particular the order of the candidates and manages the voters’ register and the ballot box. The voter performs the election action to cast their vote.

During the operation of the TOE, security-relevant events are logged by the TOE. The audit records are stored by the TOE in such a way that they are protected against unauthorized manipulation and can be reviewed by the election board at any time.

In the preparation phase, the election data, i.e. data required for the correct execution of the election, is created, corrected if necessary, and approved by the election board. The administrator is provided by the election organizer. (In the following, the election organizer appears as a user of the TOE only in the role of

the administrator. The other organizational tasks and specifications of the election organizer regarding the election, which are independent of the direct use of the TOE, are not affected by this). Each voter, the election board and the administrator have their identification data and authentication credentials available. The TOE's functionality relies on sufficient (i.e. in accordance with the specifications of the election organizer), reliable and unambiguous authentication of its users.

The installation and configuration of the TOE are performed by the administrator at the beginning of the preparation phase and have been completed successfully before the election execution starts. The TOE supports the election board in the import process of the election data.

Once the election period start time is reached, the TOE starts the execution phase. At the start of the election, the TOE ensures that the intermediate ballot box (if re-voting is allowed) and the ballot box are empty and that a self-test is performed, during which the correct functioning of the TOE is tested.

Only during the election execution, a voter is able to perform their individual election action. Votes can only be cast and finally stored in the ballot box from voters who are authorized to vote. The TOE provides the required means to the voter, or the terminal device on behalf of the voter, to enable them to encrypt their vote. The TOE processes the votes only encrypted until they are stored in a ballot box. Before stored in a ballot box, the TOE removes every link between a vote and the associated voter that might enable anyone besides the voter to connect them with their unencrypted vote.

It is not possible to change votes stored in the ballot box. In addition, votes can be read from the ballot box only for the purpose of verifying the cast-as-intended principle until the evaluation phase has been reached.

The TOE can be configured such that cast votes can be corrected, i.e. re-voting is allowed. Voters are then enabled to correct their voting decision by casting their vote again, replacing the vote previously cast by the voter. As long as votes can be corrected, they are stored in an intermediate ballot box and assigned to the voter such that the ballot secrecy is preserved. Votes that can be corrected shall not be stored in the ballot box. By means of end-to-end verification, a voter can ascertain the correct registration of their vote (recorded-as-cast).

During the execution phase, a resumption can be performed on the TOE by the election board only after a successful self-test, in case of malfunctions or crashes. The election board may also verify the correct operation of the TOE at any time by performing a self-test. The election organizer must determine under what conditions a resumption or further self-tests are to be performed by the election board.

If the election board wishes to terminate the election before the end of the execution phase, as specified by the election organizer, this will result in a suitable confirmation notice. After the end of the execution phase or the termination of the election execution, a resumption or any other form of return to the election execution is no longer possible.

After the election execution ends, the execution phase ends and the evaluation phase begins. The TOE provides the accumulated audit records to the election board, who has to check them for irregularities and afterwards has to initiate the vote count. Before starting the counting of votes, all votes in the intermediate ballot box are transferred to the ballot box. By counting all votes stored in the ballot box, the number of invalid votes and the number of valid votes are determined. By counting all valid votes, and taking into account the set of election rules for cast votes, the distribution of votes for the individual candidates shall be determined. With the determination of the result of the vote count, the election execution data and the election result are made available by the TOE in such a way that they can be stored in a manner protected from subsequent manipulation, i.e. unauthorized modifications outside the control of the TOE. In addition, the stored data enables users outside the TOE to perform end-to-end verifiability, that is cast as intended, recorded as cast, counted as recorded, individual verifiability and universal verifiability.

After the evaluation phase ends, the post-processing phase begins where the TOE can export the prepared election execution data.

The process flow for each voter's individual election action must adhere to the following principles:

- At the latest at the time of voting, the voter has been identified and authenticated.
- The voter can abort their election action at any time (until the vote is cast) without losing their voting authorization. Even in the event of a technically induced abortion, for example due to the passage of time or errors in communication, the voter's eligibility to vote must be retained. In this case, the vote must not be included in the election result.
- There is a feedback from the TOE to the voter that their vote has been successfully cast, i.e. stored in the intermediate ballot box (if re-voting is allowed) or the ballot box.
- By storing the vote in the intermediate ballot box (if re-voting is allowed) or the ballot box, the casting of the vote is recorded in the voter’s voting record.

In Figure 2, two possible processes of an electoral action are illustrated as examples:

1. The user accesses the TOE with the terminal device and opens the voting action. The TOE prompts the user to log in. The TOE checks the user's voting authorization. In the next step, the ballot is displayed to the user identified and authenticated as a voter ; all other users are rejected by the TOE.

   The voters (authorized to vote) can fill in their ballot, change it as often as they wish and make their voting decision by initiating voting. The voter is then shown their vote again. They now have the option to cast the vote in encrypted form or to revoke the initiation of voting in order to correct the vote. After successful casting of the vote, i.e. successful storage of the vote by the TOE in the intermediate ballot box (or in the ballot box if re-voting is not allowed), and the associated recording of the casting of the vote, the voters receive feedback that their vote has been successfully stored.

2. The user accesses the TOE with the terminal device and starts the voting procedure. The ballot is displayed to the user. They can fill in their ballot, change it as often as they like and make their election decision by initiating the voting process. The voter now has the option of casting the vote in encrypted form or revoking the initiation of voting in order to correct the vote.

   In the next step, the user identifies and authenticates themselves to the TOE. The TOE checks the user's voting authorization. An authorized user, i.e. a voter, is allowed to vote, all other users are rejected by the TOE. After successful voting, i.e. storage of the vote by the TOE in the intermediate ballot box (or in the ballot box if re-voting is not allowed), and the associated recording of the vote, the voter receives feedback that their vote has been successfully stored.

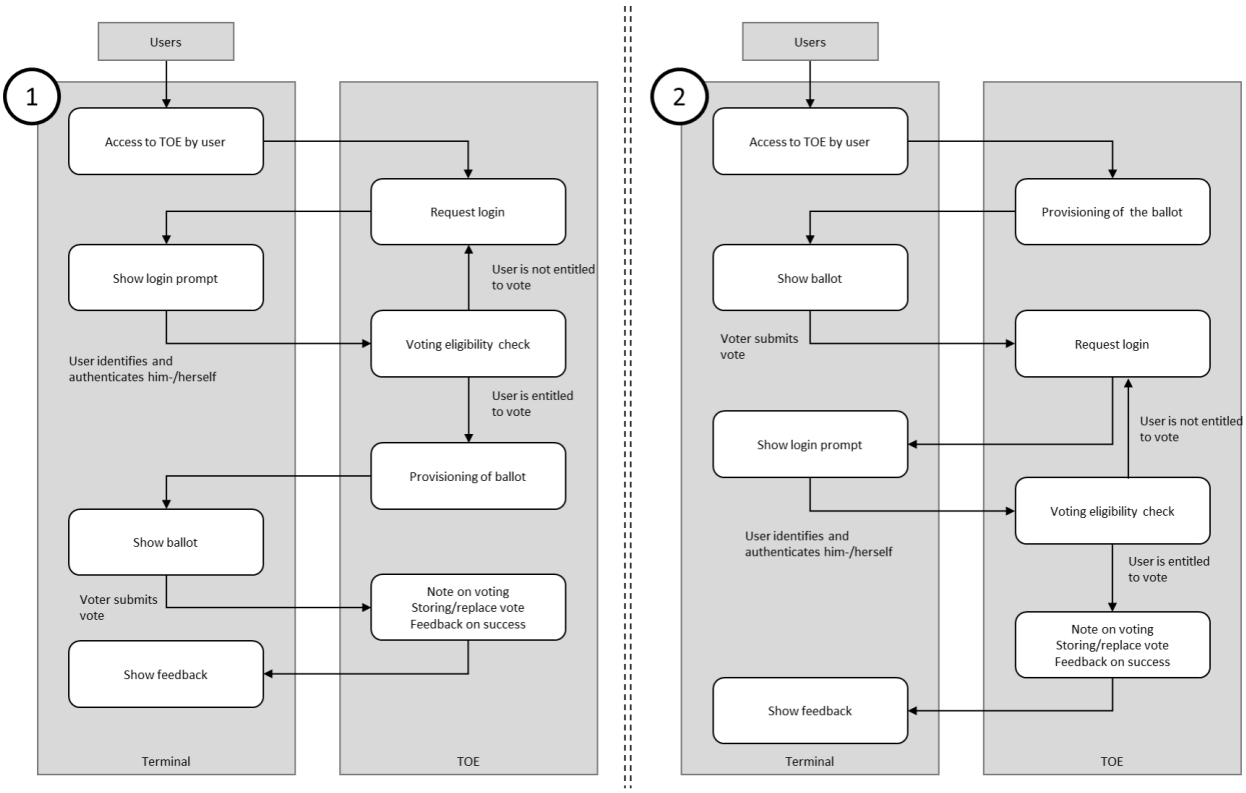

Figure 2: Exemplary processes of an election action

1.3.3 Non-TOE Hardware/Software/Firmware

The TOE is software and runs on an IT system. It therefore relies on the protection of the underlying system platform, e.g. the operating system.

The TOE's IT environment includes the parts of e-voting servers that are necessary to use the TOE, such as the hardware, operating system, a local area network in a data center environment, and an external time server.

The TOE is operated on one server which is accessible via a connected network (Internet, VPN, etc.). The network connecting the remote endpoints (including terminal devices) and the e-voting server is assumed to be any wide/local area network (WAN/LAN) without specific performance characteristics.

The terminal device used to display voting information to the voter and to initiate the election action is not part of the TOE. It must be able to display the entire content of the login screen, the ballot and the feedback, and to realize the election organizer's specifications for the type of display, in particular the order of the candidates. This can be for example a web browser, an app, or a dedicated voting device.

1.4 Terms and Definitions

Audit records: An audit record is an individual entry in an audit log related to an audited event. The audit records contain security-critical events that give information about the TOE's internal state.

Authentication credentials: Credentials that each user has to authenticate themself on the TOE. This is, for example, an individual password or a signature card on which their private key is stored.

Authentication data: Data stored in the TOE against which it is checked whether the ostensible identity of the user is genuine. These are, for example, the hash value of the password or the public key of the *voter*. The data may be stored in the *voters' register* or elsewhere.

Ballot: The list of *candidates* and other information required to correctly display the *candidates* and other specified information by the *terminal device*. The *ballot* can be blank or filled in.

Ballot box: Storage of the *cast votes,* where the cast votes cannot be changed. The *ballot box* is part of the TOE.

Ballot secrecy: In a secret election, ballot secrecy means that the *voter's* choice cannot be observed and no information about a voter's voting decision - apart from what the *election result*, after publication, reveals - can be reconstructed.

Candidate: Candidates are the items on the *candidate list*.

Candidate list: List of candidates or other choices available for selection by the *voter* on the *ballot*.

Cast vote: A *vote* is considered a cast vote, if stored in the *ballot box* in an unalterable form, or in the *intermediate ballot box* (if re-voting is allowed) in an encrypted form. The cast vote is a voter's distinct vote that will be considered for the *election result*.

Display data of the ballot: All data required to display the *ballot* correctly. If, for example, the ballot is to be displayed barrier-free by the *terminal device*, the data required for this is also summarized under this.

E-voting: E-voting**s** are elections that enable the electoral process to be carried out in electronic form, from the casting of *votes* to the counting and announcement of the *election results*, via internet or other communication networks.

E-voting server: IT systems on which the TOE is installed and to which the ***remote endpoint*** connects.

Election action: Includes identification and authentication with voting eligibility verification, ballot filling, correction and initiation of voting, display of the *vote*, casting or recasting of the vote, and *feedback* to the *voter*.

Election board: This includes both the persons who have organizational responsibility for and manage the e-voting, as well as all "vicarious agents" (e.g. employees of an e-voting service provider commissioned with the processing) who, on behalf of and under the control of senior personell of the *election board*, carry out the administration of the e-voting servers, initiate a resumption, end the election process, start the *post-processing phase* and start the counting of votes with determination of the *election result*.

Election data: Data created outside the TOE that is imported in the *preparation phase* containing data needed to define election parameters:

- The blank *ballot*, i.e. the display data of the ballot,
- the *set of election rules*,
- the *voters' register* with *authentication data*,
- the *candidate list*,
- the *election dates.*

Election dates: The dates defining the *election period start time, election period end time* and *end of election* time.

Election execution: Period in the *execution phase* during which *voters* are able to perform their individual *election action* at the TOE. The *election board* can start, resume in case of interruptions or a shutdown of the system, and terminate the election execution during the execution phase.

Election execution data: Data used and generated in the *electoral phases*, required to be exported in the *post-processing phase* and stored in a tamper-proof manner after the election. The election execution data or parts of it can be used as *verification data*. The election execution data consists of:

- *election data*,
- *voting records,*
- contents of the *ballot box*,
- *election result*,

- *audit records,* and
- *verification data.*

Election organizer: Group of people hosting the *e-voting*.

Election period: Period in the *execution phase* during which a *voter* may open an *election action* at the TOE. If the end of the election period is before the *end of election*, an election action already opened during the election period may be continued (in accordance with the respective *electoral regulation*) after the end of the election period.

Election period end time: End date defining the *election period.*

Election period start time: Starting date defining the *election period.*

Election result: The output of the *vote evaluation* is the result of the voting process on the TOE. (In the context of an *e-voting*, the *election result* can be further evaluated). The determination of the election result is based on the *set of election rules* and includes the number of invalid *votes*, the number of valid votes, and the distribution of valid votes among the individual entries of the *candidate list*, as specified by the *set of election rules*.

Electoral regulation: The electoral regulation defines the regulatory framework for conducting an election. In the context of this Protection Profile the following aspects are relevant:

- minimum number of *election board* members required to authorize security-relevant actions,
- time frame for continuing *election actions* already opened during the *election period* after the end and after the termination of the election period,
- definitions for the *set of election rules*.

End of election: The end of election is reached when the *election board* has finished the possibility of conducting the election at the TOE and marks the end of the *execution phase*. After that, no *voter* can open an *election action* at the TOE anymore and no *vote* can be sent to the TOE by the voter.

End-to-end verifiability: End-to-end verifiability can be decomposed into subrequirements:

- Cast as Intended: by casting a *vote*, the *voter's* intention was captured,
- Recorded as Cast: the *cast vote* was correctly recorded and saved,
- Counted as Recorded: the recorded vote was correctly included in the *election result*,
- Individual Verifiability: the *voter* can verify that their *vote* was counted correctly and
- Universal Verifiability: anyone can verify that the *election results* are correct.

Evaluation phase: During the evaluation phase the TOE counts the *cast votes* and determines the *election result* according to the *set of election rules*.

Execution phase: During the execution phase, *voters* can perform their individual *election action*. The *election board* can start and stop the *election execution* during the *election period.*

Feedback: The *voter* receives adequate *feedback* about the permission or refusal and the success or failure of their vote. The TOE sends a message to the voter's *terminal device* informing them accordingly, for example displaying it on the screen.

Identification data: Personal data with which users indicate their identity to the TOE. This can be, for example, a membership number, a username or a certificate ID. The term also includes the *voter*-related data in the *voters' register*, with which a voter can be uniquely identified.

Intermediate ballot box: Storage of the cast *votes* in an ecrypted form with the possibility of modification, upstream of the *ballot box*. Unlike the ballot box, the intermediate ballot box allows to have a vote linked to

the *voter* to allow re-voting. A technically conditioned intermediate storage during the transmission does not belong to it. The intermediate ballot box is part of the TOE.

Post-processing phase: During the post-processing phase the TOE can export *election execution data* to enable end-to-end verifiability after the election.

Preparation phase: During the preparation phase *election board* members can import election data into the TOE and set paremeters used in following *electoral phases*.

Remote endpoint: The IT system from which the user connects to the TOE, e.g. a browser on a PC or an App on a mobile device.

Resumption: A resumption of an election is the continuation of an *election period* that has been interrupted (e.g. due to a technical malfunction) while retaining all *cast votes* so far.

Set of election rules: Set of rules that defines:

- The validity and invalidity of *cast votes* (examples of invalid votes are that the *voter* did not select any *candidates* or selected too many) and
- the rules for assigning a vote to the *candidate* (e.g. by assigning individual weights to the *votes* of different *voters*).

Terminal device: A *remote endpoint* on which the *election action* is performed.

Verification data: Data used for the verification process of the individual and universal verifiability, see *end-to-end verifiability*. The verification data can contain enrypted votes with corresponding key material if this does not violate the *ballot secrecy*.

Vote: Content of a filled *ballot* that expresses a *voter's* will, i.e. a voter's decision to vote. This can be both a valid and an invalid *vote*. The (in)validity of a vote is defined by the *set of election rules*.

Vote evaluation: Counting of the *cast votes* in the *ballot box*. The number of invalid as well as valid votes is determined. The distribution of votes for the individual *candidate* is determined by taking the *set of election rules* for cast votes into account while counting all valid votes.

Voter: User who has successfully identified and authenticated themself to the TOE by means of a *terminal device*, who is on the *voters' register* and who is eligible to cast a *vote* at least once.

Voters' register: List of all *voters* who are allowed to participate in an election according to the *electoral regulation* applicable and considering the "universal" principle. The electoral eligibility list may additionally include the *authentication data* if not stored on an external entity.

Voting: The *voter's* consent to the storage of their *vote* in the TOE's *intermediate ballot box* (if re-voting is allowed) or *ballot box*. Voting is successful if the vote has been stored in the TOE correctly.

Voting record: The labeling of a *voter* on their successful *voting*. It is a security attribute belonging to the voter and can be stored in the *voters' register* or in another place.

2 Conformance Claim

2.1 CC Conformance Claim

This PP claims conformance to Common Criteria Version CC:2022 Revision 1 (CC:2022)

- CC Part 2 extended with FCS_ECO-EXT.
- CC Part 3 conformant

2.2 PP Claim and Package Claim

This PP does not claim conformance to any other PP.

This PP claims to be EAL 4 augmented with ALC_FLR.2.

2.3 PP Conformance Statement

This PP requires strict conformance of any PP or ST claiming conformance to it.

3 Security Problem Definition

3.1 Introduction

This section first describes which assets the TOE shall protect, which external entities interact with it, and which objects are of importance. On this basis, it then describes which threats the TOE shall defend against, which organizational security policies shall be observed, and which assumptions can be made about its operational environment.

3.1.1 Assets

The assets of the TOE are the information that the TOE is primarily used to protect. These assets are:

- The **voters' register**, whose integrity, authenticity, and confidentiality shall be protected,
- the **candidate list**, whose integrity and authenticity shall be protected,
- the **display data of the ballot**, whose integrity and authenticity shall be protected,
- the **vote**, whose integrity, authenticity and confidentiality shall be protected,

  (The confidentiality of the vote shall be understood in the context of the ballot secrecy. Even partial information about a single vote, e.g. a percentage probability for the vote or about the assignment to a voter violates the ballot secrecy, if not obtained by the election result.)
- the **verification data**, whose integrity and authenticity shall be protected,
- the **election result**, whose integrity and authenticity shall be protected,
- the **election dates**, consisting of the **election period start time**, **election period end time** and **end of election** time, whose integrity and authenticity shall be protected,
- the **set of election rules**, whose integrity and authenticity shall be protected, and
- the **election execution data,** whose integrity and authenticity shall be protected.

3.1.2 Users and Subjects

The e-voting system for non-political elections interacts with users using remote endpoints. During interaction with the TOE, the user is associated with at least one of the following roles:

- the role **unauthenticated user** has any user of the TOE who is not identified and/or authenticated,
- the role **voter** allows an authorized user to exercise their right to vote,
  (A voter is unauthorized if they already cast a vote and re-voting is not allowed.)
- the role **election board** allows the authenticated user to manage the TOE from an organizational point of view (provision and monitoring of election-related data), to check the logging data and the correct functionality of the TOE, and
- the role **administrator** allows the authenticated user to manage the TOE from a technical point of view (installation and monitoring of the technical operational environment of the TOE).

3.1.3 Security Attributes

The security attributes of users known to the TOE are

- User Identity (User-ID),
- Role determining the access rights.

Each voter has a security attribute „**voting record**“, which is false on creation and true after the voter successfully cast a vote.

Certain controlled operations performed by the election board have to be independently authorized by a minimum number of users with an election board Role. These operations are called security-critical actions and have the security attribute: **required authorisations for the operation**, which tracks how many distinct election board members authorised the operation. The initial value of the security attribute is “undefined” and has to be initiated in the preparation phase to the required number of authorisations for the actions.

While not a security attribute by itself, the TSF data item electoral phase determines the current rules for access of all subjects to any objects based on the aforementioned security attributes.The election execution data carry the security attribute “**exported”**, which is false on creation and true after successful export by the election board.

Data which can be exported, carry the security attribute “**election execution ID**”, which is an identifier to associate the exported data with the corresponding election.

3.2 Threats

T.AuthenticityTOE

An attacker redirects the user to a fake TOE without the user noticing it. The user subsequently communicates with an inauthentic TOE, causing to be violated:

- the integrity, authenticity, and confidentiality of the voters’ register,
- the integrity and authenticity of the candidate list,
- the integrity, authenticity, and confidentiality of the vote,
- the integrity and authenticity of the election dates, or
- the integrity and authenticity and of the set of election rules.

T.ContestableElection

An attacker who has access to election execution data after it has been exported falsifies or alters it so that, upon subsequent review, the correctness of the election and its election result can be illegitimately disputed.

T.ExternalCommunication

An attacker gains unauthorized access to payload data exchanged between TOE and a remote endpoint to

- change or manipulate the display data of the ballot so that the voting decision and thus the vote of the voters is influenced,
- read, exchange or manipulate parts of the votes, or unauthorized cast votes, so that the voters’ will is not represented by cast votes,
- change or manipulate the voters’ register so that users gain unauthorized access to the voting process or authorized voters are prevented from voting,
- change or manipulate the verification data,
- change or manipulate the candidate list so that the voter’s election decision is influenced, or
- modify the election dates so that authorized voters cannot exercise their right to vote, or the legal framework, defined by the election organizer, has not been complied with and the election may therefore be declared invalid.

T.Session

A user or attacker uses another authenticated user's open session on a remote endpoint to gain unauthorized access to

- read, change, or manipulate the voters' register,
- change or manipulate the candidate list,
- change or manipulate the display data of the ballot,
- change or manipulate the election dates,
- change or manipulate the set of election rules,
- read, change, or manipulate the cast votes.

T.UserData

A user or attacker gains unauthorized access to the data stored in the TOE to

- change or manipulate the election execution data,
- read the voters' register, or
- read, change, or manipulate the votes or parts of them.

Furthermore, a user or attacker casts unauthorized votes.

T.Disruption

A user or attacker disrupts the regular operation of the TOE during the election period, so that authorized voters cannot exercise their right to vote or the legal framework, defined by the election organizer, has not been complied with and the election may therefore be declared invalid.

T.Assignment

A user or attacker uses data stored in the TOE to violate the confidentiality of votes to assign it to a voter.

3.3 Organizational Security Policies

OSP.Abort

At any time before the vote is finally cast, the TOE shall offer the voter the opportunity to terminate their election action without losing eligibility to vote.

OSP.Archiving

Election execution data may only be deleted after successful export by authorized election board.

OSP.Audit

The TOE shall support the auditing of security-relevant events during operational election preparation, the election period, and post-electoral follow-up, by providing audit records. Throughout the election process, the election board may view the audit records. These must be stored on the e-voting servers protected against manipulation.

OSP.Result

The TOE must not perform the vote evaluation until the post-processing phase. The election board must review the audit records before the post-processing, including vote counting, can begin. The TOE uses a set of election rules to define valid and invalid votes and to evaluate the votes for the election result.

OSP.EmptyBallotBox

There must be no votes in the intermediate ballot box (if re-voting is allowed) and the ballot box at the beginning of the election period.

OSP.Feedback

After voting, the voter must receive a feedback about the success of their vote.

OSP.Malfunction

The election board shall be able to detect before the start and resume of the election and upon manual request by performing a self-test on the TOE, if there is a technical failure of the integrity of the TOE security functionality (TSF) or of the user and TSF data that endangers the correct operation of the TOE. After a TOE crashes or shuts down, or after a failure of communications or storage media, in a way that voting related data is unaffected, the election board shall be able to resume the election execution. In doing so, the TOE shall ensure the integrity of the election execution data.

OSP.ElectionEnd

An election action may only be opened during the election period. It must be possible to continue election actions that have already been opened after the end of the election period, in accordance with the relevant electoral regulation. It must also be possible to terminate the election ahead of time.

OSP.VotingPrinciples

The TOE provides voting and verification processes to the eligible voter during the execution phase. It enables the conduct of an e-voting in which the six election principles of “universal”, “direct”, “free”, “secret”, “equal”, and “public” are implemented according to the electoral regulation.

OSP.ElectionBoard

The operations

- to import the ballot,
- to import the voters’ register,
- to import the set of election rules,
- to import the election dates
- to terminate the election execution
- to resume the election execution
- to start the counting of votes with determination of the election result, and
- to export election execution data

may not be executed until they have been independently authorized by the minimum number of election board members required by the relevant electoral regulation.

OSP.TimeService

The TOE uses a reliable external time server. All operations of the TOE that are based on a time stamp use this time information that is independent of the operating system.

3.4 Assumptions

A.Observation

The voter is able to cast their vote unobserved. The election organizer is responsible for providing the voter with adequate instructions for unobserved voting.

A.AuthData

The users of the TOE have received all the data required to interact with the TOE, in particular the identification data and the authentication data. The users do not disclose them to other persons.

A.RemoteEndpoints

The remote endpoint allow users to verify the authenticity of the TOE. For terminal devices the following applies as well:
The voter is responsible for securing the terminal device. It is assumed that the process of voting is not observed or influenced by the terminal device. This includes that the voter does not intentionally manipulate their terminal device for such purposes. The terminal device is capable of displaying the ballot correctly, transmitting the voter's entries correctly and in encrypted form to the TOE, and deleting data that allow conclusions to be drawn about the vote cast after the election action. Every eligible voter has a terminal device that fulfills the aforementioned properties.

Note 1: Due to the election principle “universal”, it may be necessary for the ballot to be presented in an accessible manner. By A.RemoteEndpoints it is assumed that the terminal device is able to display the ballot barrier-free in such cases.

A.Network

The protection of the servers on which the TOE runs is ensured by the implementation of a security concept for the network connection. In addition, sufficient quality of service and availability of the network are provided.

A.Server

The servers running the TOE are free of malicious software that may affect the security functions of the TOE. All software on the servers is trusted and has been properly installed and updated. Unauthorized access to TOE functions, processes, and data is prevented by the servers' security mechanisms. In addition, the servers are protected against unauthorized physical access and physical manipulation.

A.AuthServer

Depending on the configuration of the remote endpoint, the user or the remote endpoint on behalf of the user checks that the remote endpoint is communicating with the correct TOE before interacting with the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

O.IdentAuth

The TOE identifies users and verifies the user's identity before granting access to controlled resources. The TOE authenticates users, and detects and responds to failed authentication attempts.

O.RuleSet

The TOE checks each received vote in the ballot box for compliance with the rules defined in the set of election rules when counting votes and determining the election result. The TOE shall incorporate each valid vote into the election result with the rules defined in the set of election rules.

O.AccessControl

The TOE provides access control to security services, operations with user data, and management of the TOE and TSF data based on the role and identity of the authenticated user and on the electoral phase.

O.TrustedChannel

The TOE provides trusted channels that use secure cryptographic mechanisms for communication between the TOE and users. The TOE ensures the authenticity, confidentiality, and integrity of the communication data exchanged over the trusted channel. In addition, the TOE provides authentication credentials to the remote endpoints that allow a user to verify the authenticity of the TOE.

O.Selftest

The TOE performs a selftest

- at the TOE‘s start phase,
- before the start of the election,
- at the request of the administrators or election boards,
- before a resumption of the election, and
- regularly in specified time intervals.

The TOE notifies administrators and election board when the selftest fails.

O.Archiving

The TOE shall generate evidence of the integrity and authenticity of election execution data, and shall enable any person to verify such evidence and attribute the data to the associated election. The TOE provides the evidence and data in such a way that it can be exported upon request by the election board. Only after successful export, the election execution data stored in the TOE can be deleted by election boards.

O.Audit

The TOE allows verification of security-related functionalities at any time by detecting, recording, and reliably and securely storing selected verifiable events using audit records related to the activities controlled by the TOE. If attacks are suspected by the TOE or if the voting process is interrupted, this will be recorded in the audit records. The audit records can be accessed and read by the election board.

O.ElectionBoard

The TOE ensures that security-critical actions of the election board are performed only if the minimum number of election board members required by the electoral regulation give their consent. The security-critical actions include importing the ballot, the set of election rules, the voters’ register, and the election

dates, as well as termination and resumption of the election execution, starting the vote count with determination of the election result, and exporting election execution data.

O.ElectionPeriod

The TOE shall ensure that no election action can be opened before the beginning, after the end and after the termination of the election period. Continuation of election actions already opened during the election period, is possible in accordance with the respective electoral regulation after the election period has been completed. The TOE enables the election board to terminate and resume the election execution during the election period.

O.BallotBox

The TOE ensures that no votes are stored in the intermediate ballot box (if re-voting is allowed) and the ballot box at the start of the election. During the election period, the ballot box can be read out only for the purpose of verifying the cast-as-intended and recorded-as-cast principles. After the end of the election period, the ballot box can be read out for determining the election result. After the cast votes or parts of them have been read out, no further vote can be cast.

O.BallotSecrecy

The TOE ensures that the ballot secrecy is maintained. To this end, the data stored on the TOE does not allow any conclusions to be drawn about the voter's vote, especially no information about a voter's voting decision - apart from what the election result, after publication, reveals - can be reconstructed.

O.SessionLimit

The TOE ensures that unused or expired sessions are automatically terminated. Futhermore users can end their sessions.

O.TimeService

The TOE uses clock synchronization provided by an external time server for providing reliable time stamps.

O.Vote

The TOE allows the voter to cast a vote and to verify the cast-as-intended, recorded-as-cast, counted-as-recorded principles for the cast vote. For this purpose, the voter receives feedback on the successful storage of their vote in the intermediate ballot box (if re-voting is allowed) or in the ballot box. With providing the feedback, the TOE stores verification data to provide the means for individual and universal verifiability. The TOE ensures that only a maximum of one cast vote exists for each voter. Before the vote is cast, the TOE allows the voter to terminate their election action without losing eligibility to cast a vote.

4.2 Security Objectives for the Operational Environment

OE.Audit

The election board or an authorized user appointed by the election board checks the audit records of the TOE at the beginning of the evaluation phase. In particular, they look for indications of attacks that suggest that election execution has been interrupted or that manipulation or unauthorized access attempts have taken place.

OE.ElectionData

The authentication credentials are distributed to all eligible voters (e.g. by the election organizer) before the election period, according to the protection needs of the election, so that only the voter with voting authorization has the respective authentication credentials at their disposal. The eligible voter does not disclose their authentication credentials or pass it on.

OE.Network

The servers running the TOE are protected against attacks from the network by implementing a security concept for the network connection. A sufficient quality of service and the availability of the network are given.

OE.Observation

The voter shall take care that no one observes them while voting. The election organizer shall provide the voter with appropriate instructions for unobserved voting.

OE.Personnel

The personnel authorized for technical and organizational administration of the TOE, i.e. the election board and the administrator, are trustworthy and instructed in the correct handling of the TOE. Specifically, the election board and administrator will close connection sessions between the endpoint and the TOE while not actively interacting with the TOE.
The administrator will not intentionally misuse the TSF.

OE.Regulations

The election board ensures that the “universal”, “direct”, “equal”, and “free” principles of the electoral regulations are fulfilled.

OE.Server

The administrators ensure that the servers running the TOE are free of malicious software that can compromise the security functions of the TOE. The servers' security mechanisms prevent unauthorized access by external software to TOE functions, processes, and data. In addition, the server is protected against unauthorized physical access and physical tampering.

OE.RemoteEndpoint

The remote endpoint allows the authenticity of the TOE to be verified by the user as specified. For terminal devices used for voting, the following applies as well:
The terminal device is capable of correctly displaying the ballot, correctly transmitting the voter's input to the TOE, performing the necessary cryptographic mechanisms to encrypt the vote locally, and deleting data that allows inference of the vote cast after the voting action. In addition, the user ensures that the terminal device with which the TOE communicates does not observe or influence the ballot casting process.

OE.TimeService

A reliable NTS-based time service is available to the TOE.

4.3 Security Objectives Rationale

The following table enables an assignment of

1. security objectives of the TOE to
   a. Threats, which are averted, and
   b. Organizational security policies that enforce security objectives,
2. security objectives of the operating environment to
   a. Threats which are averted,
   b. Organizational security policies that enforce the security objectives and
   c. Assumptions by which the security objectives are met.

Table 1: Mapping of security objectives to threats and organizational security policies

	T.UserData	T.Session	T.ExternalCommunication	T.Assignment	T.AuthenticityTOE	T.Disruption	T.ContestableElection	OSP.VotingPrinciples	OSP.Audit	OSP.Archiving	OSP.EmptyBallotBox	OSP.ElectionEnd	OSP.Feedback	OSP.Malfunction	OSP.ElectionBoard	OSP.Result	OSP.Abort	OSP.TimeService
O.Archiving							X	X	X	X								
O.Audit						X			X									
O.IdentAuth	X																	
O.TrustedChannel			X		X													
O.Selftest						X								X				
O.SessionLimit		X																
O.Vote	X							X					X				X	
O.BallotBox											X					X		
O.BallotSecrecy				X				X										
O.ElectionBoard															X			
O.ElectionPeriod						X						X		X				
O.TimeService																		X
O.AccessControl	X		X						X									
O.RuleSet																X		
OE.Audit						X										X		
OE.Observation								X										
OE.Network						X												
OE.Personnel	X	X												X				
OE.Server				X														
OE.RemoteEndpoint					X													
OE.ElectionData	X																	
OE.TimeService																		X
OE.Regulations								X										

Table 2: Mapping of security objectives for the environment to assumptions

	A.Observation	A.AuthData	A.RemoteEndpoints	A.Network	A.Server	A.AuthServer
OE.Audit						
OE.Observation	X					
OE.Network				X		
OE.Personnel		X			X	
OE.Server					X	
OE.RemoteEndpoint			X			X
OE.ElectionData		X				
OE.TimeService						
OE.Regulations						

The following describes that the security objectives counter all threats and enforce all OSPs, and that the security objectives for the operational environment maintain all assumptions.

T.UserData

O.AccessControl prohibits unauthorized users from read and write access to user data, which specifically includes the election execution data (and herein the voters' register, the candidate list, the display data of the ballot, the election result, the election dates, the set of election rules, and the verification data) and the votes. *O.AccessControl* also prohibits unauthorized users from casting the vote. Further, *O.Vote* ensures that there is a maximum of one cast vote for each voter, i.e. that the voter cannot cast additional votes.

For this purpose, *O.IdentAuth* provides the TSF the authentication mechanisms to verify credentials. The authentication credentials, as required by *OE.ElectionData*, are distributed to all eligible voters prior to the start of the election period, in accordance with the protection needs of the election, and are not shared by voters so that only the eligible voters knows their credentials.
OE.Personnel requires that the Administrator will not intentionally abuse the TSF. Since according to *OE.Personnel* the Administrator and the election board are appropriately instructed, they will not disclose their credentials to third parties.

T.Session

O.SessionLimit allows users to terminate their sessions and requires the TSF to automatically terminate unused or expired sessions. Specifically, sessions are terminated according to the requirements of *OE.Personnel* when personnel leave the remote endpoint.

T.ExternalCommunication

O.TrustedChannel ensures that data sent over the communication channels between the TOE and a remote endpoint is transmitted over trusted channels in an integrity and authenticity protected manner and cannot be read during transmission. *O.AccessControl* ensures that for casting votes these communication channels can only be used by voters.

T.Assignment

External access to the servers to gain unauthorized access to data stored in the TOE is prevented by *OE.Server*. Furthermore *O.BallotSecrecy* ensures that the secrecy of the ballot is maintained and that the data

stored and output on the TOE does not allow the assignment of voters to cast votes, even after the election results have been determined.

T.AuthenticityTOE
O.TrustedChannel ensures that the TOE provides the means to authenticate itself to the remote endpoint (including terminal devices). *OE.RemoteEndpoint* requires that, depending on the configuration of the remote endpoint, the user or the remote endpoint on behalf of the user verifies the authenticity of the TOE.

T.Disruption
O.Selftest ensures that the TOE is in a state of integrity. *O.Audit* requires that interruptions in election execution and indications of detected or suspected attacks that indicate an interruption in election execution are logged. *OE.Audit* ensures that the logged data is reviewed by knowledgeable personnel; and *O.ElectionPeriod* allows the election board to resume the election process in the event of a detected disruption. *OE.Network* ensures that the network connection to the TOE is sufficiently available during the election period.

T.ContestableElection
O.Archiving ensures that election execution data can be exported in an integrity-protected and authenticated manner.

OSP.VotingPrinciples
O.Vote enables each voter to cast their vote and to verify the casting of the vote. *OE.Observation* allows the voter to vote “freely” and “secretly”. The “secret” voting is further implemented by the TOE through *O.BallotSecrecy*. *O.RuleSet* implements the election’s “equal” principle, *O.Archiving* implements the election’s “public” principle and *OE.Regulations* implements the election’s “universal”, “equal”, “free”, and “direct” principles.

OSP.Audit
O.Audit ensures that the required logging of security-relevant events takes place and can be verified at any time. *O.AccessControl* ensures that the data and events to be logged can only be defined by the election organizer. Also *O.AccessControl* prevents the audit records from being modified before archiving. *O.Archiving* ensures that the TSF can export the election execution data in an integrity-protected manner.

OSP.Archiving
O.Archiving ensures that election execution data can be deleted only after successful export by the election board.

OSP.EmptyBallotBox
O.BallotBox implements the required functionalities directly.

OSP.ElectionEnd
O.ElectionPeriod implements the required functionalities directly.

OSP.Feedback
O.Vote implements the requested functionalities directly.

OSP.Malfunction
O.Selftest requires that a selftest be started at the required times, specifically checking the integrity of the election execution data. *OE.Personnel* ensures that the election board is trained in such a way that they can correctly classify the results of the selftest. *O.ElectionPeriod* requires that the TSFs enable the resumption of the election execution.

OSP.ElectionBoard
O.ElectionBoard implements the required functionalities directly.

OSP.Result
O.BallotBox requires that the ballot box cannot be read and the election result determined until after the

election period has ended. *OE.Audit* ensures that audit data is audited prior to the start of the post-processing. *O.RuleSet* requires that votes are evaluated according to the defined set of election rules.

OSP.Abort
O.Vote implements the required functionalities directly.

OSP.TimeService
OE.TimeService ensures that a secured time server is available and *O.TimeService* requires that the TOE uses this time server.

A.Observation
OE.Observation corresponds directly to the assumption.

A.AuthData
OE.ElectionData ensures that the assumption is implemented for the voter, and *OE.Personnel* requires the administrators and election boards to not pass on their authentication credentials.

A.RemoteEndpoints
OE.RemoteEndpoint corresponds directly to the assumption.

A.Network

OE.Network corresponds directly to the assumption.

A.Server

OE.Server corresponds to the assumption for the hardware directly, after correctly handled by the administrator as required by *OE.Personnel*, e.g. installing the software on the servers in a secure way. The administrators will furthermore not misuse the TSF intentionally.

A.AuthServer

OE.RemoteEndpoint allows the user to verify the authentication credentials of the TOE.

5 Extended Component Definition

5.1 External cryptographic Operation (FCS_ECO-EXT)

5.1.1 Family Behaviour

For a cryptographic operation to function correctly, the operation shall be performed in accordance with a specified algorithm and with a cryptographic key of a specified size. This family should be included whenever there are requirements for cryptographic operations to be performed on a TOE external entity for which the TOE provides the cryptographic algorithms.

Typical cryptographic operations include data encryption and/or decryption, digital signature generation and/or verification, cryptographic checksum generation for integrity and/or verification of checksum, secure hash (message digest), cryptographic key encryption and/or decryption, and cryptographic key agreement.

5.1.2 Components Leveling and Description

Figure 3 shows the component levelling for this family.

Figure 3: FCS_ECO-EXT Component leveling

FCS_ECO-EXT.1 Cryptographic operation, requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.

5.1.3 Management of FCS_ECO-EXT.1

The following actions can be considered for the management functions in FCS:

a) there are no management activities foreseen.

5.1.4 Audit of FCS_ECO-EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) minimal: Success and failure, and the type of cryptographic operation;

b) basic: Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.

5.1.5 FCS_ECO-EXT.1 External cryptographic Operation

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies

FCS_ECO-EXT.1.1

The TSF shall provide the code for [assignment: list of cryptographic operations] to an external entity. The cryptographic operations are in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

6 Security Requirements

The SFR components stated in this section are tailored using permitted operations:

- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list;
- Refinement: allows the addition of details; and
- Iteration: allows a component to be used more than once with varying operations.

The tailoring through assignment, selection and refinement operations is explicitly identified in each SFR component. Tailoring phrases are distinguished by the blue font color.

The tailoring through iteration operations is explicitly identified in each iterated SFR component by unique identifiers after the short name of the SFR component separated by a slash.

6.1 Security Functional Requirements (SFRs)

6.1.1 User Identification and Authentication

6.1.1.1 FMT_SMR.1 Security roles

Component relationships

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification

FMT_SMR.1.1

The TSF shall maintain the roles administrator, election board, voter, [assignment: additional authorized identified roles] [1].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.1.1.2 FIA_ATD.1 User attribute definition

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: user identity, role [2].

[1] [assignment: the authorized identified roles]
[2] [assignment: *list of security attributes*]

6.1.1.3 FIA_USB.1 User-subject binding

Component relationships

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: user identity, role [3].

FIA_USB.1.2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: the initial role of the user is “unauthenticated user” [4].

FIA_USB.1.3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: After successful identification and authentication of the user, the value of the security attribute “role” of the subject shall be set to the associated value [5].

6.1.1.4 FIA_UID.1 Timing of identification

Component relationships

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1

The TSF shall allow

- identification and authentication of the TOE to the user,
- [assignment: list of other TSF-mediated actions]. [6]

on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.1.5 FIA_UAU.1 Timing of authentication

Component relationships

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

[3] [assignment: *list of security attributes*]
[4] [assignment: *rules for the initial association of attributes*]
[5] [assignment: *rules for the changing of attributes*]
[6] [assignment: list of TSF mediated actions]

FIA_UAU.1.1

The TSF shall allow

- identification and authentication of the TOE to the user,
- identification of the user to the TOE,
- [assignment: list of other TSF-mediated actions]. [7]

on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note 1: The PP does not define any requirements regarding the cryptographic strength of authentication tokens since it is possible that an external authentication server handles authentication credentials. In any case, the ST author shall extend the PP by their own modelling of the user authentication procedures and the handling of authentication credentials. For example, authentication credentials may be stored in the TOE or handled by an external authentication server. If stored in the TOE, the ST author shall list the authentication credentials either as assets (in section 3.1.1) or as security attribute for the user (in section 3.1.3). If handled by an external authentication server, the ST author shall list the authentication server as non-TOE hardware (in section 1.3.3) and extend the security problem definition (see section 3) accordingly (e.g. considering the required interfaces and possible new threats). However, the ST author must comply with the requirements listed in the latest version of BSI-TR03107 [2] at the time of the evaluation.

6.1.1.6 FIA_AFL.1 Authentication failure handling

Component relationships

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication

FIA_AFL.1.1

The TSF shall detect when [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]] unsuccessful authentication attempts occur related to [assignment: list of authentication events].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [selection: *met, surpassed*], the TSF shall generate an audit record and [assignment: list of additional actions] [8].

6.1.2 Access Control

6.1.2.1 FDP_ACC.1 Subset access control

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute-based access control

[7] [assignment: list of TSF mediated actions]
[8] [assignment: list of actions]

FDP_ACC.1.1

The TSF shall enforce the Access Control Policy [9] on

- subjects: Administrator, election board, voter, [assignment: other roles];
- objects: election actions, vote, cast votes, voters' register, verification data, candidate list, set of election rules, display data of the ballot, election dates, election period end time, election execution data, audit records;
- operations: read, verify, open, continue, import, export, modify, delete, cast. [10]

6.1.2.2 FDP_ACF.1 Security attribute based access control

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1

The TSF shall enforce the Access Control Policy [11] to objects based on the following:

- subjects: authenticated users (attribute: "role"; for voter the additional attribute: "voting record");
- objects: election actions, vote, cast votes, voters' register, verification data, candidate list, set of election rules, display data of the ballot, election dates, election period end time, election result, the election execution data (attribute: "exported"), audit records (attribute: "exported") - for all these objects: (attribute: "electoral phase");
- operation: read, verify, open, continue, import, export, modify, delete, cast. [12]

FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

1) Authenticated users with role "voter" are allowed to open election actions, if the electoral phase is set to "execution phase" and the election period end time has not been reached.
2) Authenticated users with role "voter" are allowed to continue already opened election actions, if the electoral phase is set to "execution phase".
3) Authenticated users with role "voter" are allowed to read the following user data, if the electoral phase is set to "execution phase":
    a. candidate list
    b. display data of the ballot;
    c. election dates
    d. [selection: voters' register, set of election rules, none]
4) Authenticated users with role "voter" are allowed to verify their own cast vote, if the electoral phase is set to "execution phase".

[9] [assignment: access control SFP]
[10] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].
[11] [assignment: access control SFP]
[12] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

5) Authenticated users with role “voter” are allowed to read their own individual verification data, if the electoral phase is set to “execution phase”.

6) Authenticated users with role “election board” are allowed to read out the cast votes, as long as the electoral phase is set to “post-processing phase”.

7) Authenticated users with role “election board” are allowed to read out the verification data, as long as the electoral phase is set to “execution phase”, “evaluation phase” and “post-processing phase”.

8) Authenticated users with role “election board” are allowed to import (according to FDP_ITC.1) the following user data, after the value for required authorisations for the operation defined according to FMT_MSA.1 is reached and if the electoral phase is set to “preparation phase”:

    e. voters’ register

    f. candidate list

    g. display data of the ballot;

    h. set of election rules,.

    i. election dates

9) Authenticated users with role “election board” are allowed to read the following user data:

    j. voters’ register

    k. candidate list

    l. display data of the ballot;

    m. set of election rules,

    n. election dates

    o. audit records

10) Authenticated users with role “election board” are allowed to read the following user data, after the required authorisations for the operation is reached and if the electoral phase is set to “post-processing phase”:

    p. election result

11) Authenticated users with role “election board” are allowed to export the election execution data (according to FDP_ETC.2), if the electoral phase is set to “post-processing phase”.

12) [assignment: additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]. [13]

FDP_ACF.1.3

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

1) The election execution data shall not be deleted unless the security attribute “exported” is set to “true”, the deletion is initiated by a user with role “election board” and the required authorisations for the operation is reached.

2) No user shall read or modify the data stored in the intermediate ballot box (if re-voting is allowed) and ballot box, if the TSF is in the secure state of FPT_FLS.1/REC or FPT_FLS.1/UREC and the electoral phase is set to “execution phase”.

[13] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].

3) No user shall open or continue an election action as soon as the cast votes or parts of the cast votes have been read out.

4) No user with role “voter” shall cast a vote, if their security attribute “voting record” is set to “true” and re-voting in FMT_MOF.1 is disabled.

5) No user shall import, read, modify, export the objects listed in FDP_ACC.1, if not explicitly allowed by FDP_ACF.1.2 or FDP_ACF.1.3.

6) [assignment: additional rules, based on security attributes, that explicitly deny access of subjects to objects]. [14]

Application Note 2: Since casting a vote is considered an election action, FDP_ACF.1.2 1) and 2) also apply to cast a vote.

6.1.2.3 FDP_IFC.1/ACC Subset information flow control – Access Control

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/ACC

The TSF shall enforce the Access Control IFP [15] on the following list of subjects, information, and operations:

- subjects: authenticated user;
- information: votes, election execution data, security-critical action;
- operations: receive, decrypt, terminate, perform, initiate, decrease. [16]

6.1.2.4 FDP_IFF.1/ACC Simple security attributes – Access Control

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/ACC

The TSF shall enforce the Access Control IFP [17] based on the following types of subject and information security attributes:

- role,
- electoral phase,
- required authorisations for the operation,
- exported,
- user identity. [18]

[14] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]
[15] [assignment: information flow control SFP]
[16] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP].
[17] [assignment: information flow control SFP]
[18] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]

FDP_IFF.1.2/ACC

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

1) A vote is received via FTP_PRO.1/EP only if the electoral phase is set to execution phase, the vote is encrypted and the user transmitting the vote to the TOE is authenticated as voter.
2) An encrypted vote shall only be decrypted if it is directly saved to or already stored in the ballot box.
3) A security-critical action is performed only after the required authorisations for the operation are reached. [19]

FDP_IFF.1.3/ACC

The TSF shall enforce the following SFP rules

regarding the security attribute „required authorisations for the operation“:

- After initiating a security-critical action, the security-critical action is performed only after the required authorisations for the operation is reached, by authorisation of mutually distinct election board members.
- After the security-critical action is performed, or the required authorisations for the operation is reached, but the security-critical action may not be allowed to be performed, the track of already performed authorizations is reset.
- After initiating a security-critical action, the action may be aborted by any authenticated election board and the track of already performed authorizations is reset.

regarding the security attribute “exported”:

- After the export of election execution data by the election board, the security attribute “exported” of the respective data is set to “true”.

regarding the electoral phase:

- If the electoral phase is set to preparation phase, while the election period start time is reached and if the self test performed at the election period start time (according to FPT_TST.1) is successful, the electoral phase is set from preparation phase to execution phase.
- If the electoral phase is set to preparation phase, the electoral phase changes to execution phase, only after reaching the election period start time and after a successful self-test according to FPT_TST.1.
- After termination of the election execution, the electoral phase is set to a new phase other than the execution phase. It is set to [selection, choose one of: evaluation phase, post-processing phase, the following failure phase: assignment: [a failure phase]].
- If the electoral phase is set to execution phase, the electoral phase changes to evaluation phase as soon as the end of election time is reached,

regarding the user identity:

- Before saving the votes in the ballot box, any link to the voter’s identity is removed. [20]

FDP_IFF.1.4/ACC

The TSF shall explicitly authorise an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorise information flows].

[19] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]
[20] [assignment: additional information flow control SFP rules].

FDP_IFF.1.5/ACC

The TSF shall explicitly deny an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly deny information flows]

6.1.3 User Data Import and Export

6.1.3.1 FDP_ITC.1 Import of user data without security attributes

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation

FDP_ITC.1.1

The TSF shall enforce the Access Control Policy [21] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2

The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: additional importation control rules].

6.1.3.2 FDP_ETC.2 Export of user data with security attributes

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1

The TSF shall enforce the Access Control Policy and Access Control IFP [22] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3

The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4

The TSF shall ensure that interpretation of the security attributes of the exported user data is as intended by the owner of the user data.

[21] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[22] [assignment: access control SFP(s) and/or information flow control SFP(s)]

FDP_ETC.2.5

The TSF shall enforce the following rules when user data is exported from the TOE: [assignment: additional exportation control rules].

6.1.4 Secure Communication

6.1.4.1 FTP_PRO.1/EP Trusted channel protocol – End Points

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.2 Trusted channel establishment FTP_PRO.3 Trusted channel data protection.

FTP_PRO.1.1/EP

The TSF shall implement TLS [23] acting as TLS server [24] in accordance with: [assignment: (sub-)list of selected standards as referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [25].

FTP_PRO.1.2/EP

The TSF shall enforce usage of the trusted channel for all data exchanged between the TOE and remote endpoints [26] in accordance with: [assignment: (sub-)list of selected standards as referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [27].

FTP_PRO.1.3/EP

The TSF shall permit its peer [28] to initiate communication via the trusted channel.

FTP_PRO.1.4/EP

The TSF shall enforce the following rules for the trusted channel: [assignment: rules governing operation and use of the trusted channel and/or its protocol].

FTP_PRO.1.5/EP

The TSF shall enforce the following static protocol options: [assignment: list of options and references to standards in which each is defined].

FTP_PRO.1.6/EP

The TSF shall negotiate one of the following protocol configurations with its peer: [assignment: list of configurations and reference to standards in which each is defined].

Application Note 3: The ST author shall model all necessary trusted channel protocols by FTP_PRO.1. If different protocols are used for the voting procedure and organizational operations performed by the election board or administrator. the ST author shall iterate FTP_PRO.1.

[23] [assignment: trusted channel protocol]
[24] [assignment: defined protocol role(s)]
[25] Refinement: [assignment: list of standards]
[26] [assignment: purpose(s) of the trusted channel]
[27] Refinement: [assignment: list of standards]
[28] [selection: itself, its peer]

6.1.4.2 FTP_PRO.2/EP Trusted channel establishment – End Points

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol [FCS_CKM.1 Cryptographic key generation, or FCS_CKM.2 Cryptographic key distribution] FCS_CKM.5 Cryptographic key derivation FCS_COP.1 Cryptographic operation.

FTP_PRO.2.1/EP

The TSF shall establish a shared secret with its peer using one of the following mechanisms: [assignment: (sub-)list of selected key establishment mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [29].

FTP_PRO.2.2/EP

The TSF shall authenticate itself to its peer [30] using one of the following mechanisms: [assignment: (sub-)list of selected authentication mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [31] and according to the following rules: [assignment: list of rules for carrying out the authentication].

FTP_PRO.2.3/EP

The TSF shall use [assignment: key derivation function] to derive the following cryptographic keys from a shared secret: [assignment: list of cryptographic keys].

Application Note 4: The ST author shall list all key derivation functions used in the particular TLS standard referenced in FTP_PRO.1.

6.1.4.3 FTP_PRO.3/EP Trusted channel data protection – End Points

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol FTP_PRO.2 Trusted channel establishment FCS_COP.1 Cryptographic operation.

FTP_PRO.3.1/EP

The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [assignment: (sub-)list of selected encryption mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [32].

[29] Refinement: [assignment: list of key establishment mechanisms]
[30] [selection: its peer, itself to its peer]
[31] Refinement: [assignment: list of authentication mechanisms]
[32] Refinement: [assignment: list of encryption mechanisms]

FTP_PRO.3.2/EP

The TSF shall protect data in transit from modification, deletion, insertion, replay [33] using one of the following mechanisms: [assignment: (sub-)list of selected integrity protection mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [34].

6.1.4.4 FCS_COP.1/EP Cryptographic operation – End points

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access

FCS_COP.1.1/EP

The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [35].

6.1.4.5 FCS_CKM.1/EP Cryptographic key generation – End points

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation] FCS_CKM.3 Cryptographic key access [FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers] FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.1.1/EP

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [36].

[33] [selection: modification, deletion, insertion, replay, [assignment: other]]
[34] Refinement: [assignment: list of integrity protection mechanisms]
[35] [assignment: list of standards]
[36] [assignment: list of standards]

6.1.4.6 FCS_CKM.5/EP Cryptographic key derivation – End points

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.5.1/EP

The TSF shall derive cryptographic keys [assignment: key type] from [assignment: input parameters] in accordance with a specified key derivation algorithm [assignment: key derivation algorithm] and specified cryptographic key sizes [assignment: list of key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [37].

6.1.4.7 FCS_CKM.6/EP Timing and event of cryptographic key destruction – End points

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.6.1/EP

The TSF shall destroy [assignment: list of cryptographic keys (including keying material)] when no longer needed [38].

FCS_CKM.6.2/EP

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].

6.1.4.8 FTA_SSL.3 TSF-initiated termination

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security Roles

FTA_SSL.3.1

The TSF shall terminate an interactive session after a [assignment: *time interval of user inactivity*].

6.1.4.9 FTA_SSL.4 User-initiated termination

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

[37] [assignment: list of standards]
[38] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]

FTA_SSL.4.1

The TSF shall allow user-initiated termination of the user's own interactive session.

6.1.5 Voting and Verification

6.1.5.1 FDP_IFC.1/VOT Subset information flow control – Voting

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/VOT

The TSF shall enforce the Voting IFP [39] on the following list of subjects, information, and operations:

- subjects: voter;
- information: vote;
- operations: cast. [40]

6.1.5.2 FDP_IFF.1/VOT Simple security attributes – Voting

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/VOT

The TSF shall enforce the Voting IFP [41] based on the following types of subject and information security attributes:

- Subjects: user
- Security attributes: role, voting record, electoral phase [42]

FDP_IFF.1.2/VOT

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

A user is allowed to cast a vote only if the electoral phase is set to execution phase, the vote is encrypted before transmission from the terminal device and the user transmitting the vote to the TOE is authenticated as voter. If a user casts a vote, a previously cast vote (if existing and re-voting is allowed) is replaced by the current one. [43]

[39] [assignment: information flow control SFP]
[40] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP].
[41] [assignment: information flow control SFP]
[42] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[43] [assignment: for each operation, the security attribute-based relationship that hold between subject and information security attributes]

FDP_IFF.1.3/VOT

The TSF shall enforce the following SFP rules

regarding the security attribute „voting record“:

- After and only after successfully casting a vote by an authenticated voter, that voter's "voting record" is set to "true".
- [assignment: other additional information flow control SFP rules]. [44]

FDP_IFF.1.4/VOT

The TSF shall explicitly authorise an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorise information flows].

FDP_IFF.1.5/VOT

The TSF shall explicitly deny an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly deny information flows].

6.1.5.3 FCS_ECO-EXT.1 External cryptographic operation

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies

FCS_ECO-EXT.1.1

The TSF shall provide the code for encrypting a vote [45] to an external entity. The cryptographic operations are in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

6.1.5.4 FDP_IFC.1/VFY Subset information flow control – Verify

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/VFY

The TSF shall enforce the Verify IFP [46] on the following list of subjects, information, and operations:

- subjects: voter;
- information: cast vote, verification data;
- operations: verify the cast-as-intended principle, verify the recorded-as-cast principle, and verify the counted-as-recorded principle. [47]

[44] [assignment: additional information flow control SFP rules].
[45] [assignment: list of cryptographic operations]
[46] [assignment: information flow control SFP]
[47] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP].

6.1.5.5 FDP_IFF.1/VFY Simple security attributes – Verify

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/VFY

The TSF shall enforce the Verify IFP [48] based on the following types of subject and information security attributes:

- Subjects: user, TOE;
- Security attributes: role, electoral phase. [49]

FDP_IFF.1.2/VFY

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- The TOE shall provide feedback to the voter regarding the successful storage of the cast vote and shall store the verification data, enabling the voter to verify the cast-as-intended principle by the use of [assignment: algorithm used to implement cast-as-intended principle].
- A user authenticated as voter is allowed to verify their vote according to the recorded-as-cast principle only if the electoral phase is set to execution phase by the use of [assignment: algorithm used to implement recorded-as-cast principle].
- The TOE enables a user authenticated as voter to verify their vote according to the counted-as-recorded principle and provides information in the election execution data that enables a voter to verify the counted-as-recorded principle for their vote after the electoral phase is set to “post-processing phase” by the use of [assignment: algorithm used to implement cast-as-recorded principle]. [50]

FDP_IFF.1.3/VFY

The TSF shall enforce the [assignment: additional information flow control SFP rules].

FDP_IFF.1.4/VFY

The TSF shall explicitly authorise an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorise information flows].

FDP_IFF.1.5/VFY

The TSF shall explicitly deny an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly deny information flows]

[48] [assignment: information flow control SFP]
[49] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[50] [assignment: for each operation, the security attribute-based relationship that hold between subject and information security attributes]

6.1.5.6 FCS_COP.1/BAL Cryptographic operation – Ballot box integrity

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access

FCS_COP.1.1/BAL

The TSF shall perform integrity verification [51] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

6.1.5.7 FCS_CKM.1/BAL Cryptographic key generation – Ballot box

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation] FCS_CKM.3 Cryptographic key access [FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers] FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.1.1/BAL

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

6.1.5.8 FCS_CKM.6/BAL Timing and event of cryptographic key destruction

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.6.1/BAL

The TSF shall destroy [assignment: list of cryptographic keys (including keying material)] when no longer needed [52].

FCS_CKM.6.2/BAL

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].

[51] [assignment: list of cryptographic operations]
[52] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]

6.1.6 Electoral Evaluation according to rule sets

6.1.6.1 FDP_IFC.1/EE Subset information flow control – Electoral Evaluation

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/EE

The TSF shall enforce the Election Evaluation IFP [53] on the following list of subjects, information, and operations:

- subjects: authenticated user;
- information: cast votes, set of election rules, election result;
- operations: evaluation of validity, counting of valid and invalid cast votes, evaluation of vote distribution, determine the election result. [54]

6.1.6.2 FDP_IFF.1/EE Simple security attributes – Electoral Evaluation

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/EE

The TSF shall enforce the Election Evaluation IFP [55] based on the following types of subject and information security attributes:

- Role,
- electoral phase,
- required authorisations for the operation. [56]

FDP_IFF.1.2/EE

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- The electoral phase is set to evaluation phase.
- The operations are initiated by a user with the role “election board”.
- If the operation shall be authorized by the minimum number of election board members, the operation starts only after the required authorisations for the operation is reached. [57]

[53] [assignment: information flow control SFP]
[54] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP].
[55] [assignment: information flow control SFP]
[56] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[57] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]

FDP_IFF.1.3/EE

The TSF shall enforce the following SFP rules:

- To determine the election result, the operation shall be authorized by the minimum number of election board members.
- Before initiating any of the SFP rules mentioned below, all cast votes stored in the intermediate ballot box are transferred to the ballot box (if re-voting is allowed).
- The validity of a cast vote stored in the ballot box is evaluated according to the set of election rules.
- The distribution of votes for the individual candidates are determined according to the set of election rules and using all valid cast votes stored in the ballot box.
- [assignment: other additional information flow control SFP rules]. [58]

FDP_IFF.1.4/EE

The TSF shall explicitly authorise an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorise information flows].

FDP_IFF.1.5/EE

The TSF shall explicitly deny an information flow based on the following rules:

- If either the electoral phase is not set to evaluation phase or the operations are initiated by a user with a Role different than the election board Role, all information flow regarding the cast votes stored in the ballot box shall be denied, if not explicitly authorized in FDP_IFF.1.4/EE.
- [assignment: additional rules, based on security attributes, that explicitly deny information flows]. [59]

6.1.6.3 FDP_SDI.2 Stored data integrity monitoring and action

Component relationships

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.

FDP_SDI.2.1

The TSF shall monitor user data stored in the intermediate ballot box (if re-voting is allowed) and ballot box [60] controlled by the TSF for deletion, unauthorized changes [assignment: additional integrity errors] [61] on all objects, based on the following attributes: [assignment: user data attributes].

FDP_SDI.2.2

Upon detection of a data integrity error, the TSF shall provide an information to the election board, generate an audit record according to FAU_GEN.1 and result in a failed self-test according to FPT_TST.1 [62].

6.1.7 Audit

6.1.7.1 FAU_GEN.1 Audit data generation

Component relationships

[58] [assignment: additional information flow control SFP rules]
[59] [assignment: rules, based on security attributes, that explicitly deny information flows]
[60] Refinement: “container”
[61] [assignment: integrity errors]
[62] [assignment: action to be taken]

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps

FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:

1. Start-up and shutdown of the audit functions;
2. All auditable events for the not specified [63] level of audit; and
3. Too many unsuccessful authentication attempts defined by FIA_AFL.1;
4. Processing of election data;
5. Processing of time data;
6. Starting and ending of the execution phase;
7. Start, resumption and termination of an election;
8. Processing data of election actions;
9. Start, stop and results of self-tests;
10. Processing of election execution data;
11. Data integrity errors according to FDP_SDI.2;
12. Detection of indications of potential attacks by FAU_SAA.3;
13. Interruptions of the election execution;
14. Access to data stored in the intermediate ballot box (if re-voting is allowed) and ballot box, if not for the purpose of verifying the cast-as-intended principle;
15. The audit data storage exceeds the threshold defined by FAU_STG.4;
16. [assignment: other additionally specifically defined auditable events]. [64]

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

1. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
2. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information].

6.1.7.2 FPT_STM.1 Reliable time stamps

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps by synchronizing the internal clock with an external time server [65].

[63] [*selection: choose one of: minimum, basic, detailed, not specified*]
[64] [assignment: other specifically defined auditable events]
[65] Refinement: Added external time server

6.1.7.3 FAU_SAA.3 Simple attack heuristics

Component relationships

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_SAA.3.1

The TSF shall be able to maintain an internal representation of the following signature events

1. Brute-force attacks on identification and authentication credentials,
2. network-based DoS attacks on external TOE interfaces,
3. [assignment: other subsets of system events]. [66]

that may indicate a violation of the enforcement of the SFRs.

FAU_SAA.3.2

The TSF shall be able to compare the signature events against the record of system activity discernible from an examination of [assignment: the information to be used to determine system activity].

FAU_SAA.3.3

The TSF shall be able to indicate a potential violation of the enforcement of the SFRs when a system event is found to match a signature event that indicates a potential violation of the enforcement of the SFRs.

6.1.7.4 FAU_STG.2 Protected audit trail storage

Component relationships

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.2.1

The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.2.2

The TSF shall be able to detect [67] unauthorized modifications to the stored audit records in the audit trail.

6.1.7.5 FAU_STG.4 Action in case of possible audit data loss

Component relationships

Hierarchical to: No other components.

Dependencies: FAU_STG.2 Protected audit data storage

FAU_STG.4.1

The TSF shall

1. prevent the establishment of trusted channels for the communication with remote users, other than the [selection: election board, administrator, [assignment: other authorized role]],
2. generate an audit record according to FAU_GEN.1,

[66] [assignment: a subset of system events]
[67] [selection: choose on of: prevent, detect]

3. generate a notification by [assignment: type of notification], distinct from the audit record, for the [selection: election board, administrator, [assignment: other authorized role]], and
4. [assignment: additional actions to be taken in case of possible audit data storage failure] [68]

if the audit data storage exceeds [assignment: pre-defined limit].

6.1.7.6 FAU_SAR.1 Audit review

Component relationships

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation

FAU_SAR.1.1

The TSF shall provide the election board [69] with the capability to read all audit events [70] from the audit data.

FAU_SAR.1.2

The TSF shall provide the audit records in a manner suitable for the user to interpret the information and to determine the correctness of the election execution [71].

6.1.8 Archiving

6.1.8.1 FCS_COP.1/EXP Cryptographic operation – Export of Archiving Data

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access

FCS_COP.1.1/EXP

The TSF shall perform signature generation over exported data [72] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [73].

6.1.8.2 FCS_CKM.6/EXP Timing and event of cryptographic key destruction

Component relationships

Hierarchical to:	No other components.

[68] [assignment: actions to be taken in case of possible audit data storage failure]
[69] [assignment: authorised users]
[70] [assignment: list of audit information]
[71] Refinement: Added purpose
[72] [assignment: list of cryptographic operations]
[73] [assignment: list of standards]

Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.6.1/EXP

The TSF shall destroy [assignment: list of cryptographic keys (including keying material)] when [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]].

FCS_CKM.6.2/EXP

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].

6.1.9 Reaching and preserving secure states and resuming the process

6.1.9.1 FPT_TST.1 TSF testing

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_TST.1.1

The TSF shall run a suite of self tests

- during initial start-up,
- Reaching the election period start time,
- Before the start of the election,
- by request of the election board or the administrator,
- before the resumption of the election,
- regularly in specified time intervals,
- after manually leaving the maintenance mode from FPT_RCV.1, and
- [assignment: additional conditions under which self test should occur] [74]

to demonstrate the correct operation of the TSF [75]:

- Verify the integrity and consistency of the ballot box and intermediate ballot box (if re-voting is allowed),
- Verify the integrity, authenticity, and consistency of already imported election data,
- Verify the correct functioning of audit generation,
- Verify that the interfaces required for the establishment of the trusted channel with remote endpoints are performing as intended and enable the establishment of the trusted channels according to FTP_PRO.1/EP,
- Verify the integrity of access control mechanisms,
- Verify the integrity of the synchronization status of the internal clock, and

[74] [selection: during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self test should occur]]
[75] [selection: [assignment: parts of TSF], the TSF]

- [assignment: additional list of self-tests run by the TSF] [76].

FPT_TST.1.2

The TSF shall provide authorized users with the capability to verify the integrity of TSF data [77].

FPT_TST.1.3

The TSF shall provide authorised users with the capability to verify the integrity of TSF [78].

6.1.9.2 FPT_FLS.1/REC Failure with preservation of secure state – Recoverable

Component relationships

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1/REC

The TSF shall preserve a secure state when the following types of failures occur:

- self test (FPT_TST.1) fails recoverable,
- the intermediate ballot box (if re-voting is allowed) or the ballot box are not empty at the start of the election,
- a crash/shutdown of the system. [79]

6.1.9.3 FPT_FLS.1/UREC Failure with preservation of secure state - Unrecoverable

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1/UREC

The TSF shall preserve a secure state when the following types of failures occur:

- self test (FPT_TST.1) fails unrecoverable. [80]

6.1.9.4 FPT_RCV.1 Manual recovery

Component relationships

Hierarchical to: No other components.

Dependencies: AGD_OPE.1 Operational user guidance

FPT_RCV.1.1

After failures defined in FPT_FLS.1/REC, [assignment: additional list of failures/service discontinuities] [81] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

[76] [assignment: list of self-tests run by the TSF]
[77] [selection: [assignment: parts of TSF data], TSF data]
[78] [selection: [assignment: parts of TSF], TSF]
[79] [assignment: list of types of failures in the TSF]
[80] [assignment: list of types of failures in the TSF]
[81] [assignment: list of failures/service discontinuities]

6.1.10 Management of Security Attributes

6.1.10.1 FMT_MSA.3 Static attribute initialization

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles

FMT_MSA.3.1

The TSF shall enforce the Access Control Policy, Access Control IFP and Election Evaluation IFP [82] to provide restrictive [83] default values for security attributes that are used to enforce the SFP, i.e. the newly generated or updated election execution data shall have the security attribute "exported" set to "false", the voter's "voting record" shall be set to "false", the security attribute required authorisations for the operation for the security-critical actions is set to "undefined" [84].

FMT_MSA.3.2

The TSF shall allow [85]no-one [86] to specify alternative initial values to override the default values when an object or information is created.

6.1.10.2 FMT_MSA.1 Management of security attributes

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1

The TSF shall enforce the Access Control Policy, Access Control IFP and Election Evaluation IFP [87] to restrict the ability to

(1) Modify [88] the security attributes electoral phase [89] according to the following list [90]:

1) the election board may initiate the change from preparation phase to execution phase,
2) the election board may initiate the change from execution phase to [selection, choose one of: evaluation phase, post-processing phase, the following failure phase: assignment: [a failure phase]], after a suitable confirmation notice, and
3) the election board may initiate the change from evaluation phase to post-processing phase. [91]

[82] [assignment: access control SFP, information flow control SFP]
[83] [selection, choose one of: restrictive, permissive, [assignment: other property]]
[84] Refinement: Added examples
[85] Refinement: Deleted "the"
[86] [assignment: the authorized identified roles]
[87] [assignment: *access control SFP(s), information flow control SFP(s)*]
[88] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[89] [assignment: *list of security attributes*]
[90] Refinement: replaced "to"
[91] [assignment: *the authorized identified roles*]

(2) Change [92] the security attributes required authorisations for the operation [93] from the value "undefined" to a positive integer number [94] to [selection: administrator, election board] [95] only during the preparation phase [96].

Application Note 5: The refinements repeat parts of the SFR component to avoid iteration of the component.

Application Note 6: The initiated change from execution phase shall be considered a termination of the election execution.

6.1.11 Management of Security Functions

6.1.11.1 FMT_MTD.1 Management of TSF data

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1

The TSF shall restrict the ability to

1) define [97] the additional conditions for the self-tests (according to FPT_TST.1.1) [98] to [selection: administrator, election board]; [99]
2) define [100] the time interval for regular self-tests (according to FPT_TST.1.1) [101] to [selection: administrator, election board]; [102]
3) complement [103] the following list defining the security-critical actions [104] to [selection: administrator, election board] [105]

only to the preparation phase [106].

The list of security-critical actions consists of:

- the import of the ballot,
- the import of the voters' register,
- the import of the set of election rules,
- the import of the election dates,
- the termination of the election execution,
- the resumption of the election execution,

[92] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[93] [assignment: *list of security attributes*]
[94] Refinement: Added values
[95] [assignment: *the authorized identified roles*]
[96] Refinement: Added phase
[97] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[98] [assignment: list of TSF data]
[99] [assignment: the authorized identified roles]
[100] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[101] [assignment: list of TSF data]
[102] [assignment: the authorized identified roles]
[103] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[104] [assignment: list of TSF data]
[105] [assignment: the authorized identified roles]
[106] Refinement: Added additional restriction to phase

- the start of counting votes with determination of the election result,
- the export of election execution data, and
- [assignment: additional security critical actions]. [107]

6.1.11.2 FMT_MOF.1 Management of security functions behavior

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1

The TSF shall restrict the ability to

- enable [108] the functions leaving the maintenance mode from FPT_RCV.1 [109] to election board [110],
- disable the behaviour of [111] the functions re-voting (cast a vote if the voter's security attribute "voting record" is set to true according to FDP_ACF.1) [112] to [selection: administrator, election board]. [113]

6.1.11.3 FMT_SMF.1 Specification of Management Functions

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:

- Management of TSF data to perform the elections according to the respective electoral regulation:
  - define the additional conditions for the self-tests (according to FPT_TST.1.1),
  - define the time interval for regular self-tests (according to FPT_TST.1.1),
  - complement the list defining the security-critical actions.
- [assignment: list of additional management functions to be provided by the TSF]. [114]

6.2 Security Assurance Requirements (SARs)

The PP requires the TOE to be evaluated according to EAL 4 augmented with ALC_FLR.2, and with specific refinements on AGD_OPE.1. The (unrefined) SAR components are taken from CC Part 3 and referenced in Table 6.

Table 3: Security Assurance Requirements (SARs)

[107] Refinement: Added definition of security-critical actions
[108] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[109] [assignment: list of functions]
[110] [assignment: the authorised identified roles]
[111] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[112] [assignment: list of functions]
[113] [assignment: the authorised identified roles]
[114] [assignment: list of management functions to be provided by the TSF]

ASSURANCE CLASS	*ASSURANCE COMPONENTS: ABBREVIATION*	*ASSURANCE COMPONENTS*
ADV: DEVELOPMENT	ADV_ARC.1	Security architecture description
	ADV_FSP.4	Complete functional specification
	ADV_IMP.1	Implementation representation of the TSF
	ADV_TDS.3	Basic modular design
AGD: GUIDANCE DOCUMENTS	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: LIFE-CYCLE SUPPORT	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_FLR.2	Flaw reporting procedures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
ASE: SECURITY TARGET EVALUATION	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ATE: TESTS	ATE_COV.2	Analysis of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA: VULNERABILITY ASSESSMENT	AVA_VAN.3	Focused vulnerability analysis

Application Note 7: For AVA_VAN.3 the ST author and the evaluation facility shall consider the contents of [1] and [4] as typical concepts for vulnerability assessments (if applicable web components are used in the TOE).

6.2.1 Assurance Refinements

Refinement on AGD_OPE.1.3C – AGD_OPE.1.4C:

The operational user guidance shall, for the administrator or board member, especially describe the security-relevant events relative to the regular time intervals for self-tests (according to FPT_TST.1.1) to be performed. Examples for secure values for these time intervals shall be provided, depending on different use cases and the election period's duration. These guidelines shall ensure that the regular self-tests are performed sufficiently often during the election period.

6.3 Security Requirements Rationale

6.3.1 Justification of SFR/SAR dependencies

All dependencies of the SAR components are satisfied.

All dependencies of the SFR components are satisfied, not applicable or shall be addressed by the author of the Security Target by selecting the appropriate dependencies (see Table 4).

Table 4: Justifications for security requirements

SFR component	Dependencies	Justification
FAU_GEN.1	FPT_STM.1 Reliable time stamps	satisfied
FAU_SAA.3	No dependencies.	
FAU_SAR.1	FAU_GEN.1 Audit data generation	satisfied
FAU_STG.2	FAU_GEN.1 Audit data generation	satisfied
FAU_STG.4	FAU_STG.2 Protected audit data storage	satisfied
FCS_CKM.1/EP	FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/EP
	FCS_CKM.3 Cryptographic key access	not satisfied because it is technically not needed for the desired security functionality of the TOE in context of FCS_CKM.1/EP.
	FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers	shall be satified by Security Target author: This Protection Profile does not specify whether the source for random data is covered by FCS_RBG.1 or FCS_RNG.1 because both cases shall be covered by this Protection Profile without restricting a Security Target to one specific requirement. The Security Target author shall fulfill the dependencies according to TOE's capabilities.
	FCS_CKM.6 Timing and event of cryptographic key destruction	satisfied by FCS_CKM.6/EP
FCS_CKM.1/BAL	FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/BAL
	FCS_CKM.3 Cryptographic key access	not satisfied because it is technically not needed for the desired security functionality of the TOE in context of FCS_CKM.1/BAL.
	FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers	shall be satified by Security Target author: This Protection Profile does not specify whether the source for

SFR component	Dependencies	Justification
		random data is covered by FCS_RBG.1 or FCS_RNG.1 because both cases shall be covered by this Protection Profile without restricting a Security Target to one specific requirement. The Security Target author shall fulfill the dependencies according to TOE's capabilities.
	FCS_CKM.6 Timing and event of cryptographic key destruction	satisfied by FCS_CKM.6/BAL
FCS_CKM.6/BAL	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation	satisfied by FCS_CKM.1/BAL
FCS_CKM.5/EP	FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/EP
	FCS_CKM.6 Timing and event of cryptographic key destruction	satisfied by FCS_CKM.6/EP
FCS_CKM.6/EP	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation	satisfied by FCS_CKM.1/EP
FCS_COP.1/BAL	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_CKM.1/BAL
	FCS_CKM.3 Cryptographic key access	not satisfied because it is technically not needed for the desired security functionality of the TOE in context of FCS_COP.1/BAL.
FCS_COP.1/EXP	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation	shall be satified by Security Target author: This Protection Profile does not specify whether the cryptographic keys are imported by the TOE user into the TOE or are generated on the TOE itself because both cases shall be covered by this Protection Profile without restricting a Security Target to one specific requirement. The Security Target author shall fulfill the dependencies according to TOE's capabilities.
	FCS_CKM.3 Cryptographic key access	not satisfied because it is technically not needed for the desired security functionality of the TOE in context of FCS_COP.1/EXP.
FCS_CKM.6/EXP	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation or FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_COP.1/EXP
FCS_COP.1/EP	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_CKM.1/EP
	FCS_CKM.3 Cryptographic key access	not satisfied because it is technically not needed for the desired security functionality of

SFR component	Dependencies	Justification
		the TOE in context of FCS_COP.1/EP.
FCS_ECO-EXT.1	No dependencies.	
FDP_ACC.1	FDP_ACF.1 Security attribute-based access control	satisfied
FDP_ACF.1	FDP_ACC.1 Subset access control	satisfied
	FMT_MSA.3 Static attribute initialisation	satisfied
FDP_ETC.2	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	satisfied by FDP_ACC.1
FDP_IFC.1/ACC	FDP_IFF.1 Simple security attributes	satisfied by FDP_IFF.1/ACC
FDP_IFC.1/EE	FDP_IFF.1 Simple security attributes	satisfied by FDP_IFF.1/EE
FDP_IFC.1/VFY	FDP_IFF.1 Simple security attributes	satisfied by FDP_IFF.1/VFY
FDP_IFC.1/VOT	FDP_IFF.1 Simple security attributes	satisfied by FDP_IFF.1/VOT
FDP_IFF.1/ACC	FDP_IFC.1 Subset information flow control	satisfied by FDP_IFC.1/ACC
	FMT_MSA.3 Static attribute initialisation	satisifed
FDP_IFF.1/EE	FDP_IFC.1 Subset information flow control	satisfied by FDP_IFC.1/EE
	FMT_MSA.3 Static attribute initialisation	satisifed
FDP_IFF.1/VFY	FDP_IFC.1 Subset information flow control	satisfied by FDP_IFC.1/VFY
	FMT_MSA.3 Static attribute initialisation	satisifed
FDP_IFF.1/VOT	FDP_IFC.1 Subset information flow control	satisfied by FDP_IFC.1/VOT
	FMT_MSA.3 Static attribute initialisation	satisifed
FDP_ITC.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	satisfied by FDP_ACC.1
	FMT_MSA.3 Static attribute initialisation	satisifed
FDP_SDI.2	No dependencies.	
FIA_AFL.1	FIA_UAU.1 Timing of authentication	satisifed
FIA_ATD.1	No dependencies.	
FIA_UAU.1	FIA_UID.1 Timing of identification	satisifed
FIA_UID.1	No dependencies.	
FIA_USB.1	FIA_ATD.1 User attribute definition	satisfied
FMT_MOF.1	FMT_SMR.1 Security roles	satisfied
	FMT_SMF.1 Specification of Management Functions	satisfied
FMT_MTD.1	FMT_SMR.1 Security roles	satisfied
	FMT_SMF.1 Specification of Management Functions	satisfied
FMT_MSA.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	satisfied by FDP_ACC.1
	FMT_SMR.1 Security roles	satisfied
	FMT_SMF.1 Specification of Management Functions	satisfied
FMT_MSA.3	FMT_MSA.1 Management of security attributes	satisfied
	FMT_SMR.1 Security roles	satisfied
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1 Timing of identification	satisfied
FPT_FLS.1/REC	No dependencies.	
FPT_FLS.1/UREC	No dependencies.	
FPT_STM.1	No dependencies.	

SFR component	Dependencies	Justification
FPT_TST.1	No dependencies.	
FPT_RCV.1	AGD_OPE.1 Operational user guidance	satisfied
FTA_SSL.3	FMT_SMR.1 Security Roles	satisfied
FTA_SSL.4	No dependencies.	
FTP_PRO.1/EP	FTP_PRO.2 Trusted channel establishment	satisfied by FTP_PRO.2/EP
	FTP_PRO.3 Trusted channel data protection	satisfied by FTP_PRO.3/EP
FTP_PRO.2/EP	FTP_PRO.1 Trusted channel protocol	satisfied by FTP_PRO.1/EP
	FCS_CKM.1 Cryptographic key generation, or FCS_CKM.2 Cryptographic key distribution	satisfied byFCS_CKM.1/EP
	FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_CKM.5/EP
	FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/EP
FTP_PRO.3/EP	FTP_PRO.1 Trusted channel protocol	satisfied by FTP_PRO.1/EP
	FTP_PRO.2 Trusted channel establishment	satisfied by FTP_PRO.2/EP
	FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/EP

6.3.2 SFR Mapping to Security Objectives for the TOE

All SFR components trace to security objectives for the TOE (see Table 5).

O.Vote:

The *Voting IFP* (defined by *FDP_IFC.1/VOT* and *FDP_IFF.1/VOT*) ensures that voters can cast a vote and that at most one valid cast vote per voter exists. The *Verify IFP* (defined by *FDP_IFC.1/VFY* and *FDP_IFF.1/VFY*) ensures that voters can verify the cast-as-intended, recorded-as-cast and counted-as-recorded principles on the cast vote. The *Verify IFP* includes that the TOE provides a feedback to the voter about the successful storage of their vote in the intermediate ballot box or ballot box.

O.IdentAuth:

FIA_ATD.1 requires the TSF to maintain the list of security attributes user identity, and role from individual users. *FIA_USB.1* requires the TSF to associate each user initially with the unauthenticated user role, and only after identification and authentication associate them with their respective Role. *FIA_UID.1* requires the TSF to deny access to controlled resources before the user is identified. *FIA_UAU.1* requires that identified users need to be authenticated successfully before other TSF mediated action. *FIA_AFL.1* requires the TSF to detect and react to failed authentication attempts.

O.RuleSet:

The SFRs *FDP_IFF.1/EE* and *FDP_IFC.1/EE* define the *Election Evaluation IFP*. The *Election Evaluation IFP* requires the TSF to apply the set of election rules to determine the validity of each cast vote and to determine the election result.

Since any deletion of any cast vote stored in the ballot box is detected as required by *FDP_SDI.2*, each received vote is stored in the ballot box when the election result is determined. *FCS_COP.1/BAL* and *FCS_CKM.1/BAL* supply the required cryptographic procedures for data integrity error detection.

The *Election Evaluation IFP* furthermore requires the TSF to transfer each cast vote to the ballot box and to use each valid cast vote, stored in the ballot box, to determine the election result.

O.AccessControl:

The *Access Control Policy* is described by the SFR *FDP_ACC.1*. *FDP_ACF.1* defines the access control rules and restricts access to the user data consisting of votes, cast votes, voters' register, candidate list, set of election rules, display data of the ballot, election dates, election execution data and audit records based on the

authenticated users, their associated role, the electoral phase and the voting record in case of authenticated voter.

The requirements to import the voters' register, candidate list, display data of the ballot, set of election rules and election dates is defined by *FDP_ITC.1*. The requirements to export the election execution data is defined by *FDP_ETC.2*.

FDP_IFC.1/ACC describes the *Access Control IFP*. *FDP_IFF.1/ACC* defines, among other things, the rules for the security-critical actions, which are defined in *FMT_MTD.1*. The roles are described in *FMT_SMR.1*.

FIA_USB.1 binds authenticated users to their roles and defines the secure initial values. For security-critical actions, for voters, and election execution data*, FMT_MSA.3* defines initial values for their security attributes and *FMT_MSA.1* defines the management functions to be performed on this security attributes.

Initialization of the electoral phase is not required as this is not bound to any subject or object which may be created.

FDP_IFF.1/ACC defines the rules under which the electoral phase changes automatically how the required authorisations for the operation is reached and the election execution data's "exported" attribute is set to "true". FDP_IFF.1/VOT defines the rules under which the "voting record" changes from "false" to "true". *FMT_MSA.1* defines the rules under which the electoral phase change can be initiated manually, and *FMT_MOF.1* restricts leaving the maintenance mode to election board.

The capabilities for management of TSF data is defined by *FMT_SMF.1*.

FDP_IFF.1/EE and *FDP_IFC.1/EE* define the restrictions to evaluate the votes and determine the election result.

FMT_MTD.1 defines the management functions to perform the elections according to the respective electoral regulation.

O.TrustedChannel:

FTP_PRO.1/EP *and FTP_PRO.3*/EP require the TSF to support a trusted path to users with assured identification of its end points and protection of data from modification and disclosure. *FCS_COP.1/EP* supplies the required cryptographic procedures for data encryption/ decryption, data integrity failure detection and data authentication. The cryptographic keys for *FCS_COP.1/EP* are established using *FCS_CKM.1/EP* and *FCS_CKM.5/EP*.

After termination of the trusted path *FCS_CKM.6/EP* is used to delete these keys.

To allow the user (or the remote endpoint on behalf of the user) to verify the authenticity of the TOE, *FTP_PRO.2/EP* requires the TSF to provide verifiable authentication credentials to the remote endpoint.

O.Selftest:

FPT_TST.1 enables the election board to perform a test sequence at the TOE's start phase, before the start of the election, at the request of the administrators or election board and before a resumption of the election. It furthermore requires the TSF to perform these self-tests regularly. If such tests fail the TSF enter a secure state according to *FPT_FLS.1/REC* or *FPT_FLS.1/UREC* and an audit record is generated as required by *FAU_GEN.1*. During the execution phase it denies the access to the intermediate ballot box (if re-voting is allowed) and the ballot box by *FDP_ACF.1*, *FDP_IFF.1/ACC*, and *FDP_IFC.1/ACC* if the TSF reaches the *FPT_FLS.1/REC* secure state or *FPT_FLS.1/UREC* secure state.

O.Audit:

FAU_GEN.1 requires the TSF to generate an audit record for verifiable events and to record indications of potential attacks, indicated by attack heuristics defined by *FAU_SAA.3*. Furthermore, interruptions of the election execution are also recorded in the audit records, e.g. the interruptions due to changing to a secure state according to *FPT_FLS.1/REC*.

FAU_STG.2 and *FAU_STG.4* require the TSF to reliably and securely store the audit data to prevent loss of audit records and by notifying and forcing authorized users to act if the audit data storage is getting full.

The audit records can be accessed and reviewed by the election board as ensured by *FAU_SAR.1*.

O.Archiving:

FCS_COP.1/EXP requires the TSF to generate a signature over exported data and to enable any person to verify evidence of the integrity and authenticity of the exported election execution data. *FDP_ETC.2* requires the election execution data to carry the security attribute "election execution ID", to distinguish the data between different executed elections. Hence, it requires the TSF to provide evidence to associate the data to the executed election. Currently the PP does not state how the relevant key for *FCS_COP.1/EXP* is provided and shall be done by the ST Author. *FCS_CKM.6/EXP* destroys the relevant cryptographic material.

FDP_ACF.1 allows the election board to export the election execution data, which by *FDP_IFF.1/ACC* and *FDP_IFC.1/ACC* set the "exported" security attribute to "true", which in turn allows the election board to delete exported entries by *FDP_ACF.1*. *FMT_MSA.3* ensures that freshly generated or updated election execution data are not marked as "exported", which means they have to be exported before deletion.

O.BallotSecrecy:

FCS_ECO-EXT.1 provides the algorithm to encrypt the vote locally on the terminal device. By *FDP_IFF.1/ACC* and *FDP_IFC.1/ACC*, only encrypted votes are transmitting to the TOE using the trusted channel *FTP_PRO.1*/EP*, FTP_PRO.2*/EP *, and FTP_PRO.3*/EP. The information flow policy *FDP_IFF.1/ACC* requires the TSF to prevent the decryption of the encrypted votes as long as they are not directly saved to or already stored in the ballot box, furthermore any vote stored in the ballot box has no link to the voter's identity. *FCS_ECO-EXT.1* defines the cryptographic operations which ensure that the cast vote stored in the ballot box cannot be linked to the voter with data stored on the TOE.

O.ElectionPeriod:

FDP_ACF.1 and *FDP_IFF.1/ACC* prohibit election actions to be opened in the stated situations, while requiring the continuation of election actions already opened during the election period to be allowed until the end of election. The SFR *FMT_MTD.1* defines how the end of election may be configured as required by *FMT_SMF.1*.

The resumption of the election is ensured by *FPT_RCV.1* and restricted to the election board by *FMT_MOF.1*.

The termination of the election process is enabled by *FMT_MSA.1* with help of *FMT_SMR.1*.

The termination of the election period ends the execution phase immediately, according to *FDP_IFF.1/ACC*. The ST author shall specify if the electoral phase is set to "evaluation phase", "post-processing phase" or a failure phase. In case of a failure phase, the ST author shall define the operations which are permitted to be performed.

O.BallotBox:

By *FPT_FLS.1/REC* the TSF enters a secure state if the intermediate ballot box (if re-voting is allowed) and the ballot box are not empty at the start of the election. In this case the election cannot start, as the electoral phase cannot be changed to "execution phase", according to *FDP_IFF.1/ACC*, *FDP_IFC.1/ACC* and *FDP_ACF.1*.

To read out the intermediate ballot box (if re-voting is allowed) and ballot box, the electoral phase has to be set to "post-processing phase", according to *FDP_ACF.1*.

FDP_IFF.1/EE and *FDP_IFC.1/EE* ensure that the election result is determined after the election period during the evaluation phase.

FDP_ACF.1 prohibts to transmit a vote if cast votes or parts of them have been read out. The concrete procedure to ensure this shall be defined by the ST author.

O.ElectionBoard:

The information flow policy *Access Control IFP* (defined by *FDP_IFC.1/ACC* and *FDP_IFF.1/ACC*) requires the security attribute required authorisations for the operation to be reached by authorisation by the election board to perform the security-critical actions. The security-critical actions are defined in *FMT_MSA.1* with help of *FMT_SMR.1*.

FDP_ITC.1 defines the import of the ballot, the set of election rules and the voters' register; restaring the election process is enabled by *FPT_RCV.1*; starting the vote count with determination of the election result is enabled by *FDP_IFF.1/EE* and *FDP_IFC.1/EE*; and the export of election execution data is enabled by *FDP_ETC.2*.

O.SessionLimit:

The SFRs *FTA_SSL.3* and *FTA_SSL.4* implement the requirements directly.

O.TimeService:

The TSF *FPT_STM.1* with the refinement implements the requirements directly.

Table 5: Mapping from SFRs to objectives

SFR component	O.Archiving	O.Audit	O.IdentAuth	O.TrustedChannel	O.Selftest	O.SessionLimit	O.Vote	O.BallotBox	O.BallotSecrecy	O.ElectionBoard	O.ElectionPeriod	O.TimeService	O.AccessControl	O.RuleSet
FAU_GEN.1		X			X									
FAU_SAA.3		X												
FAU_SAR.1		X												
FAU_STG.2		X												
FAU_STG.4		X												
FCS_CKM.1/EP				X										
FCS_CKM.1/BAL														X
FCS_CKM.6/EXP	X													
FCS_CKM.5/EP				X										
FCS_CKM.6/EP				X										
FCS_COP.1/BAL														X
FCS_COP.1/EXP	X													
FCS_COP.1/EP				X										
FCS_ECO-EXT.1									X					
FDP_ACC.1													X	
FDP_ACF.1	X				X						X		X	
FDP_ETC.2	X									X			X	
FDP_IFC.1/ACC	X				X				X	X	X		X	
FDP_IFC.1/EE										X	X		X	X
FDP_IFC.1/VFY							X							
FDP_IFC.1/VOT							X							
FDP_IFF.1/ACC	X				X				X	X	X		X	
FDP_IFF.1/EE										X	X		X	X
FDP_IFF.1/VFY							X							
FDP_IFF.1/VOT							X						X	
FDP_ITC.1										X			X	
FDP_SDI.2														X

SFR component	O.Archiving	O.Audit	O.IdentAuth	O.TrustedChannel	O.Selftest	O.SessionLimit	O.Vote	O.BallotBox	O.BallotSecrecy	O.ElectionBoard	O.ElectionPeriod	O.TimeService	O.AccessControl	O.RuleSet
FIA_AFL.1			X											
FIA_ATD.1			X											
FIA_UAU.1			X											
FIA_UID.1			X											
FIA_USB.1			X										X	
FMT_MOF.1													X	
FMT_MTD.1													X	
FMT_MSA.1										X			X	
FMT_MSA.3	X												X	
FMT_SMF.1													X	
FMT_SMR.1										X	X		X	
FPT_FLS.1/REC		X			X									
FPT_FLS.1/UREC					X									
FPT_STM.1												X		
FPT_TST.1					X									
FPT_RCV.1										X				
FTA_SSL.3						X								
FTA_SSL.4						X								
FTP_PRO.1/EP				X					X					
FTP_PRO.2/EP				X					X					
FTP_PRO.3/EP				X					X					

6.3.3 Explanation of the chosen SARs

The chosen SAR package EAL4 augmented with ALC_FLR.2 provides a consistent level of rigour and assurance that is appropriate to the type of the TOE.

The augmentation ALC_FLR.2 does not add any dependencies.

7 Package for multi-component Server Architecture

7.1 Identification

Title	Multi-component Server Architecture Package
Short title	BSI-CC-PP-0121
Version	1.0
Date	2023-12-1
Sponsor	Federal Office for Information Security, Germany
Editor	Evaluation Facility of Deutsche Telekom Security GmbH
Registration	Federal Office for Information Security, Germany
Certification ID	BSI-CC-PP-0121
CC Version	CC:2022 Revision 1
Conformance Claim	CC Part 2 conformant CC Part 3 conformant

7.2 Introduction

The base Protection Profile defines that the TOE consists of one server component. This package is about a multi-component server architecture where the TOE consists of more than one server component, e.g. mix-net architectures.

This package defines trusted communication channels, over which the TOE exchanges data between the particular components.

7.2.1 TOE Type

Instead of only one server component in the base PP, the target of evaluation (TOE) is a server software consisting of more than one server components for conducting secret non-political e-votings.

7.2.2 Usage and major Security Features of the TOE

Instead of only one server component in the base PP, the TOE is a product that realizes all its functions on more than one component. Remote endpoints (not part of the TOE) access the services of the TOE remotely via a secure connection enabled by the TOE (cf. Figure 4).

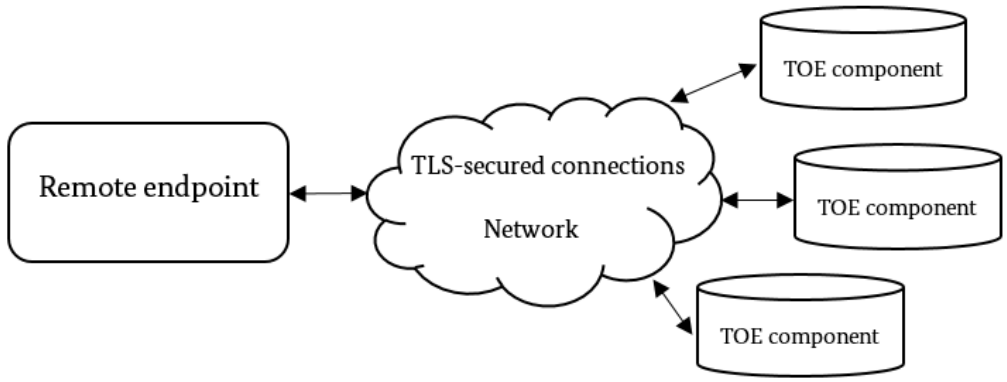

Figure 4: Multi-component TOE

7.2.3 Non-TOE Hardware/Software/Firmware

Instead of only one server component as in the base PP, the TOE is operated on more than one server, which are accessible via a connected network (Internet, VPN, etc.). The network connecting the remote endpoint and the e-voting servers is assumed to be any wide/local area network (WAN/LAN) without specific performance characteristics.

7.3 Security Problem Definition

7.3.1 Threats

This package defines additional threats, which shall be considered and mitigated because the network connection between the TOE components may be routed over TOE external network infrastructure.

T.InternalCommunication

An attacker gains unauthorized access to payload data exchanged between TOE components to

- change or manipulate the display data of the ballot so that the voting decision and thus the vote of the voters is influenced,
- read, exchange or manipulate parts of the votes, or unauthorized cast votes, so that the voters' will is not represented by cast votes,
- change or manipulate the voters' register so that users gain unauthorized access to the voting process or authorized voters are prevented from voting,
- change or manipulate the candidate list so that the voter's election decision is influenced, or
- modify the election dates so that authorized voters cannot exercise their right to vote or the legal framework, defined by the election organizer, has not been complied with and the election may therefore be declared invalid.

7.3.2 Organizational Security Policies

This package does not define any organisational security policies.

7.3.3 Assumptions

This package does not define any assumptions.

7.4 Security Objectives

7.4.1 Security Objectives for the TOE

O.TrustedInternalChannel

The TOE provides trusted channels that use secure cryptographic mechanisms for communication between the TOE components. The TOE ensures the authenticity, confidentiality, and integrity of the communication data exchanged over the trusted channel. In addition, the TOE uses mutual authentication between the TOE components that allow each component to verify the authenticity of the other TOE components.

7.4.2 Security Objectives for the Operational Environment

This package does not define any security objectives for the operational environment of the TOE.

7.4.3 Security Objectives Rationale

Table 6: Mapping of security objectives to threats and organizational security policies

	T.InternalCommunication
O.TrustedInternalChannel	X

T.InternalCommunication

O.TrustedInternalChannel ensures that data sent over the communication channels between TOE components is transmitted over trusted channels in an integrity and authenticity protected manner and cannot be read during transmission.

7.5 Security Requirements

The SFR components stated in this section are tailored using permitted operations:

- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list;
- Refinement: allows the addition of details; and
- Iteration: allows a component to be used more than once with varying operations.

The tailoring through assignment, selection and refinement operations is explicitly identified in each SFR component. Tailoring phrases are distinguished by the blue font color.

The tailoring through iteration operations is explicitly identified in each iterated SFR component by unique identifiers after the short name of the SFR component separated by a slash.

7.5.1 Security Functional Requirements (SFRs)

7.5.1.1 FTP_PRO.1/TC Trusted channel protocol – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.2 Trusted channel establishment FTP_PRO.3 Trusted channel data protection.

FTP_PRO.1.1/TC

The TSF shall implement TLS [115] acting as TLS client and server [116] in accordance with: [assignment: (sub-)list of selected standards as referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [117].

FTP_PRO.1.2/TC

The TSF shall enforce usage of the trusted channel for all data exchanged between TOE components [118] in accordance with: [assignment: (sub-)list of selected standards as referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [119].

[115] [assignment: trusted channel protocol]
[116] [assignment: defined protocol role(s)]
[117] Refinement: [assignment: list of standards]
[118] [assignment: purpose(s) of the trusted channel]
[119] Refinement: [assignment: list of standards]

FTP_PRO.1.3/TC

The TSF shall permit itself and its peer [120] to initiate communication via the trusted channel.

FTP_PRO.1.4/TC

The TSF shall enforce the following rules for the trusted channel: [assignment: rules governing operation and use of the trusted channel and/or its protocol].

FTP_PRO.1.5/TC

The TSF shall enforce the following static protocol options: [assignment: list of options and references to standards in which each is defined].

FTP_PRO.1.6/TC

The TSF shall negotiate one of the following protocol configurations with its peer: [assignment: list of configurations and reference to standards in which each is defined].

7.5.1.2 FTP_PRO.2/TC Trusted channel establishment – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol [FCS_CKM.1 Cryptographic key generation, or FCS_CKM.2 Cryptographic key distribution] FCS_CKM.5 Cryptographic key derivation FCS_COP.1 Cryptographic operation.

FTP_PRO.2.1/TC

The TSF shall establish a shared secret with its peer using one of the following mechanisms: [assignment: (sub-)list of selected key establishment mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [121].

FTP_PRO.2.2/TC

The TSF shall authenticate its peer and itself to its peer [122] using one of the following mechanisms: [assignment: (sub-)list of selected authentication mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [123] and according to the following rules: [assignment: list of rules for carrying out the authentication].

FTP_PRO.2.3/TC

The TSF shall use [assignment: key derivation function] to derive the following cryptographic keys from a shared secret: [assignment: list of cryptographic keys].

Application Note 8: The ST author shall list all key derivation functions used in the particular TLS standard referenced in FTP_PRO.1.

[120] [selection: itself, its peer]
[121] Refinement: [assignment: list of key establishment mechanisms]
[122] [selection: its peer, itself to its peer]
[123] Refinement: [assignment: list of authentication mechanisms]

7.5.1.3 FTP_PRO.3/TC Trusted channel data protection – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol FTP_PRO.2 Trusted channel establishment FCS_COP.1 Cryptographic operation.

FTP_PRO.3.1/TC

The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [assignment: (sub-)list of selected encryption mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [124].

FTP_PRO.3.2/TC

The TSF shall protect data in transit from modification, deletion, insertion, replay [125] using one of the following mechanisms: [assignment: (sub-)list of selected integrity protection mechanisms referenced in latest version of BSI TR02102-2 [3] at the time of the product evaluation] [126].

7.5.1.4 FCS_COP.1/TC Cryptographic operation – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access

FCS_COP.1.1/TC

The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [127].

7.5.1.5 FCS_CKM.1/TC Cryptographic key generation – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation] FCS_CKM.3 Cryptographic key access [FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers] FCS_CKM.6 Timing and event of cryptographic key destruction

[124] Refinement: [assignment: list of encryption mechanisms]
[125] [selection: modification, deletion, insertion, replay, [assignment: other]]
[126] Refinement: [assignment: list of integrity protection mechanisms]
[127] [assignment: list of standards]

FCS_CKM.1.1/TC

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [128].

7.5.1.6 FCS_CKM.5/TC Cryptographic key derivation – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.5.1/TC

The TSF shall derive cryptographic keys [assignment: key type] from [assignment: input parameters] in accordance with a specified key derivation algorithm [assignment: key derivation algorithm] and specified cryptographic key sizes [assignment: list of key sizes] that meet the following: latest version of BSI TR02102-2 [3] at the time of the evaluation [129].

7.5.1.7 FCS_CKM.6/TC Timing and event of cryptographic key destruction – TOE Components

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.6.1/TC

The TSF shall destroy [assignment: list of cryptographic keys (including keying material)] when no longer needed [130].

FCS_CKM.6.2/TC

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].

7.5.2 Security Requirements Rationale

7.5.2.1 Justification of SFR dependencies

All dependencies of the SFR components are satisfied, not applicable or shall be addressed by the author of the Security Target by selecting the appropriate dependencies (see Table 7).

[128] [assignment: list of standards]
[129] [assignment: list of standards]
[130] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]

Table 7: Justifications for security requirements

SFR component	Dependencies	Justification
FCS_CKM.1/TC	FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/TC
	FCS_CKM.3 Cryptographic key access	Not satisfied because it is technically not needed for the desired security functionality of the TOE in the context of FCS_CKM.1/TC
	FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers	shall be satified by Security Target author: This Protection Profile does not specify whether the source for random data is covered by FCS_RBG.1 or FCS_RNG.1 because both cases shall be covered by this Protection Profile without restricting a Security Target to one specific requirement. The Security Target author shall fulfill the dependencies according to TOE's capabilities.
	FCS_CKM.6 Timing and event of cryptographic key destruction	satisfied by FCS_CKM.6/TC
FCS_CKM.5/TC	FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/TC
	FCS_CKM.6 Timing and event of cryptographic key destruction	satisfied by FCS_CKM.6/TC
FCS_CKM.6/TC	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation	satisfied by FCS_CKM.1/TC
FCS_COP.1/TC	FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_CKM.1/TC
	FCS_CKM.3 Cryptographic key access	Not satisfied because it is technically not needed for the desired security functionality of the TOE in the context of FCS_COP.1/TC
FTP_PRO.1/TC	FTP_PRO.2 Trusted channel establishment	satisfied by FTP_PRO.2/TC
	FTP_PRO.3 Trusted channel data protection	satisfied by FTP_PRO.3/TC
FTP_PRO.2/TC	FTP_PRO.1 Trusted channel protocol	satisfied by FTP_PRO.1/TC
	FCS_CKM.1 Cryptographic key generation, or FCS_CKM.2 Cryptographic key distribution	satisfied byFCS_CKM.1/TC
	FCS_CKM.5 Cryptographic key derivation	satisfied by FCS_CKM.5/TC
	FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/TC
FTP_PRO.3/TC	FTP_PRO.1 Trusted channel protocol	satisfied by FTP_PRO.1/TC
	FTP_PRO.2 Trusted channel establishment	satisfied by FTP_PRO.2/TC
	FCS_COP.1 Cryptographic operation	satisfied by FCS_COP.1/TC

7.5.2.2 SFR Mapping to Security Objectives for the TOE

O.TrustedInternalChannel

FTP_PRO.1/TC and *FTP_PRO.3/TC* require the TSF to support a trusted path between TOE components with assured identification of its end points and protection of data from modification and disclosure. *FCS_COP.1/TC* supplies the required cryptographic procedures for data encryption/ decryption, data integrity failure detection and data authentication. The cryptographic keys for *FCS_COP.1/TC* are established using *FCS_CKM.1/TC* and *FCS_CKM.5/TC*.

After termination of the trusted path *FCS_CKM.6/TC* is used to delete these keys.

To allow the user (or the remote endpoint on behalf of the user) to verify the authenticity of the TOE, *FTP_PRO.2/TC* requires the TSF to provide verifiable authentication credentials to the remote endpoint.

Table 8: Mapping of security objectives to threats and organizational security policies

SFR component	O.TrustedInternalChannel
FCS_CKM.1/TC	X
FCS_CKM.5/TC	X
FCS_CKM.6/TC	X
FCS_COP.1/TC	X
FTP_PRO.1/TC	X
FTP_PRO.2/TC	X
FTP_PRO.3/TC	X

8 Bibliography

[1] Ergebnisse des Projektes MaSiOWa, Markt- und Sicherheitsanalyse von Online-Wahlprodukten, Federal Office for Information Security, version 1.0, 2022-05-10, URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03169/MaSiOWa.html

[2] BSI TR03107-1 Elektronische Identitäten und Vertrauensdienste im E-Government, Federal Office for Information Security, latest version, URL: https://www.bsi.bund.de/dok/TR-03107

[3] BSI TR02102-2 Kryptografische Verfahren: Verwendung von Transport Layer Security (TLS), latest version, URL: https://www.bsi.bund.de/dok/TR-02102

[4] OWASP Web Security Testing Guide, OWASP Foundation, latest version, URL: https://owasp.org/www-project-web-security-testing-guide/stable/

[5] BSI TR03162 IT-sicherheitstechnische Anforderungen zur Durchführung einer Online-Wahl im Rahmen des Modellprojektes nach § 194a Fünftes Buch Sozialgesetzbuch (Online-Wahl), latest version, URL: https://www.bsi.bund.de/dok/TR-03162

[6] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, CC:2022 Revision 1, 2022-11
Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, CC:2022 Revision 1, 2022-11
Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, CC:2022 Revision 1, 2022-11
Common Criteria for Information Technology Security Evaluation – Part 4: Framework for the specification of evaluation methods and activities, CC:2022 Revision 1, 2022-11
Common Criteria for Information Technology Security Evaluation – Part 5: Pre-defined packages of security requirements, CC:2022 Revision 1, 2022-11
Common Criteria for Information Technology Security Evaluation – Evaluation methodology, CC:2022 Revision 1, 2022-11