Melanie Volkamer

LNBIP 30

Evaluation of Electronic Voting

Requirements and Evaluation Procedures to Support Responsible Election Authorities

Springer

Lecture Notes
in Business Information Processing 30

Melanie Volkamer

Evaluation of Electronic Voting

Requirements and Evaluation Procedures to Support Responsible Election Authorities

Springer

Author

Melanie Volkamer
Technische Universität Darmstadt
Center for Advanced Security Research Darmstadt
Mornewegstraße 32, 64293 Darmstadt, Germany
e-mail: volkamer@cased.de

Vom Promotionsausschuss des Fachbereichs 4: Informatik
der Universität Koblenz-Landau zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Dr.rer.nat)
genehmigte Dissertation

Vorsitzender des Promotionsausschusses:
Prof. Dr. Dieter Zöbel

Vorsitzender der Promotionskommission:
Prof. Dr. Dieter Zöbel

Berichterstatter:
Prof. Dr. Rüdiger Grimm
Prof. Dr. Jörg Schwenk (Ruhr-Universität, Bochum)

Datum der wissenschaftlichen Aussprache:
27. Oktober 2008

Library of Congress Control Number: Applied for

ACM Computing Classification (1998): J.1, K.4, D.4.6, D.2.1

ISSN 1865-1348
ISBN-10 3-642-01661-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01661-5 Springer Berlin Heidelberg New York

springer.com

Typesetting: Camera-ready by author, data conversion by Markus Richter, Heidelberg
Printed on acid-free paper SPIN: 12677061 06/3180 5 4 3 2 1 0

Foreword

Electronic voting has a young, but attractive history, both in the design of basic cryptographic methods and protocols, and in the application of communities who wanted to be in the vanguard of new technologies. Even before 2000, when ICANN was elected Internet-wide by e-mail, voting machines were used in many political elections all over the world, almost without public notice or criticism. Perhaps only the conflicts over the Bush–Gore polls in Florida in 2000 made the public aware of the risks of electronic voting machines. Since then, diverging opinions on the use of information technology for voting, especially on the Internet, have been expressed loudly. In 2007 the Internet was highly accepted by the voting public in Estonia. On the other hand, in 2008 the attacks on the NEDAP voting machines and the High Court procedures in The Netherlands, Ireland and Germany shed a lot of stage light on electronic voting.

At the same time, the scientific community has brought forward an extensive understanding of the risks and prospects of electronic voting. In particular, IT security is subject to research not only by computer scientists, but also by legal, social and political scientists. One of the most urgent questions is the suitable expression of security of voting systems; more precisely: how can security requirements on voting systems be specified, how can such requirements be evaluated, how can products be evaluated to match specified requirements?

Only a few works in the recent past have investigated these questions as elaborately as this book by Melanie Volkamer on the "Evaluation of Electronic Voting," which specifies "Requirements and Evaluation Procedures to Support Responsible Election Authorities." It is not only a scientifically sound piece of research work, which has received a distinguished mark by my supervising university. It is also a highly educational text for anybody who wants to understand what electronic voting is about and how it can be made more secure. The reader learns that electronic voting has two branches: one branch comprises electronic voting machines in traditional precincts, which are only used for more effective counting; the other branch deals with Internet voting (also called remote electronic voting) to replace the paper ballots and their delivery via postal voting or physical ballot boxes. The latter seems to be more risky, more vanguard, more influential on the political atmosphere. The reader is introduced to both forms of electronic voting by concrete examples of their usage. The reader learns how the security of information technology is measured and evaluated in general. In a didactically well-organized way, the book introduces the international standardized methodology of the Common Criteria, and especially the part of the Common Criteria that deals with protection profiles of user requirements.

In accordance with the German Society of Computer Scientists (Gesellschaft für Informatik e.V.), Melanie Volkamer has succeeded in specifying a "Protection Profile for a Basic Set of Security Requirements for Online Voting Products." This profile was certified by the German Federal Office for Security in Information Technology (Bundesamt für Sicherheit in der Informationstechnik - BSI) in May 2008. This book discusses the reasons that this profile was created. While promoting the protection profile, the Gesellschaft für Informatik and other communities have introduced Internet voting in their election procedures. This has stimulated the development of real products, as well as the continuous observation of the products and their usage. Thus, the book is a result of this live debate in the public. The findings of this book are, therefore, not only based on theoretical considerations, but also on a related practical experience.

It is worthwhile to emphasize two more highlights (out of many) of this book. It follows the perspective of the Common Criteria especially in the important distinction between the system to be evaluated and the trust environment in which such a system is used. The technical aspects of a secure system are associated with its application environment, including human and organizational aspects. One of the findings of Melanie Volkamer is that the trust model of a voting system needs to be evaluated against the degree of the separation of duty principle. The separation of duty principle requires that risky operations are carried out by more than one person, who are unlikely to collaborate. Therefore, systems need a certain robustness against the danger of unauthorized collaboration. The method of the Common Criteria has no means to express this kind of robustness. To overcome this, Melanie Volkamer suggests a "k-resilience" meassure. This approach is important not only for electronic voting, but for other Internet applications as well.

Another highlight is the introduction of formal modelling. Of course, the exact method of formal modelling is not of very wide public interest, but its effect is important for everybody. Consistency and freedom of contradictions can only be ensured by formal methods. And which application in the world is more sensitive for a sound democratic life than voting? If voting does not require the highest security standard, which one does? This book takes a first step into security modelling of voting and shows how it might be continued.

I am convinced that this book deals with one of the hottest topics of IT applications and that it addresses the most urgent aspects of electronic voting.

The readers will not only learn a lot more about voting, and Internet voting in particular, but they will be stimulated in their own research and development work. I wish the book and its readers much success in this direction.

February 2009 Rüdiger Grimm

Preface

First, I would like to gratefully acknowledge my supervisor Prof. Dr. Rüdiger Grimm, who continuously helped me during this work with his support, his motivating words and his constantly quick reactions to my numerous e-mails—particularly in the last couple of months. Second, I am indebted to Prof. Dr. Jörg Schwenk for being my second supervisor.

Furthermore, I would like to thank the many people who encouraged me with technical support: Roland Vogt and Marcel Weinand for the technical discussions on the application of the Common Criteria in general but also in detailed aspects, Dieter Hutter for the very helpful comments on the formal IT security model, and Margaret McGaley for the helpful discussions during the requirement definitions for voting machines.

I am grateful to the PhD students of the e-Voting seminar group for numerous stimulating discussions and general advice; in particular I would like to acknowledge the help of Robert Krimmer for his support and the great collaboration, which I really enjoyed.

I would like to thank my former employer—the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI - German Research Center of Artificial Intelligence)—in particular my boss Prof. Dr. Jörg Siekmann, who enabled me to continue my research on electronic voting. In addition, I would like to thank my colleague Dieter Hutter for introducing me to the IT security community. I thank Prof. Dr. Hermann De Meer and Prof. Dr. Dirk Heckmann from the University of Passau, who gave me the opportunity to finish my thesis, to meet my supervisor, to publish papers, and to attend several conferences. In addition, I would like to thank the dean of the faculty for computer science of the University of Koblenz-Landau Prof. Dr. Dieter Zöbel and his assistant Brigitte Fuhrich for their understanding and support. I would also like to give a special thanks to the European Science Foundation for making my research visit at the National University of Ireland - Maynooth possible and thereby the collaboration with Margaret McGaley and Prof. Dr. Paul Gibson. I want to thank the Deutscher Akademischer Austauschdienst (DAAD - German Academic Exchange Service) that made my research visit at the

Technical University of Eindhoven possible and helped me to set up a great collaboration with Berry Schoenmakers, Hugo Jonker, and Prof. Dr. Bart Jacobs. I am very grateful to the five above-mentioned fantastic researchers for the discussions and their hospitality.

I would like to thank the group leaders of various electronic voting initiatives that invited me to join their groups, which made it possible for me to discuss different system approaches and different evaluation techniques.

I would like to thank Michael Kreutzer and Georg Rock for their recommendations and encouraging words, particularly in the last phase of my PhD thesis.

I would like to thank Robert, Hugo, Margaret, Jessica, Andrea, Lothar, and Jörg for their support in proof-reading my thesis, especially in terms of English wording.

Finally, I am indebted to my parents, my sister and brother as well as Axel for their understanding, endless patience, and their support. I am also grateful to Ammar and Robert as very good friends for their encouragement when it was most required. They played a decisive role in keeping me working on this thesis.

March 2009 Melanie Volkamer

Contents

Part I Fundamentals

Part II Requirements

Part III Evaluation

Part IV Application

Part V Conclusion

Part VI Appendix

1 Introduction

1.1 Elections and Electronic Voting

Election History. The history of elections goes back a long way and election reforms have taken place several times: People in the ancient world wrote their decisions on shards, Englishmen in the Middle Ages cast their vote by public acclamation, Italians in the Renaissance used either polling bowls or polling stones, and voters in Prussia wrote their decisions in public electoral registers (compare to [42]). Starting at the end of the 19th century many, especially industrial, countries established the following election reforms: first, the principles of a *universal franchise* as well as a *direct* and *secret election* were introduced. Later, the principles of *free* and *equal suffrage* were added leading to the fact that women and poor people received the right to vote. In Germany all five principles have been explicitly embedded in the law since 1949 (compare to [122]). We might say that this can be considered as the beginning of modern democracy.

More recently, several countries all over the world have introduced further reforms: Postal voting and voting in advanced polling stations have been used to strengthen the universal franchise, though they weaken the secrecy of the vote and voter freedom. In Germany postal voting was integrated in the law in 1956, in Switzerland in 1991 (compare to [16]), and in Austria in 2008. Mechanical and later electronic voting machines were introduced to save money and time. In Germany, mechanical voting machines were integrated in the law in 1975 and electronic ones in 1999 (compare to [143]). In the U.S. the first mechanical lever machines were already in use in 1892 in New York and by the 1930's in all larger urban centres. The U.S. revolution of electronic voting began in the 1960's with the introduction of punch-card ballots, continuing with the introduction of optical mark-sense ballots and direct recording electronic (DRE) voting machines in the 1970's (compare to [131] for the American voting machine history).

In the beginning, electronic voting was limited to the use of electronic voting machines. Nowadays, we are more and more faced with the next major

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 1–10, 2009.

challenge: The introduction of remote electronic voting. This would enable voters to cast their vote over the Internet from any place and any device capable of connecting to the Internet (popular examples where remote electronic voting systems have already been used are Estonia and Switzerland).

Elections on Different Political Levels. With respect to governmental elections you can distinguish between local, provincial or national elections, and referenda. In addition, there exists many non-governmental elections: Examples include elections in companies, public authorities, organisations, societies, clubs, and associations, at universities and schools, as well as award nominations and general opinion surveys.

All these types of elections differ in their importance, degree of interest for manipulations, environment, and number of voters. Although the election principles apply to all of them, they are subject to different electoral laws. For instance, some elections on lower political levels have less stringent requirements with respect to the secrecy of the vote than governmental elections.

The general agreement in many countries is to gain experience with remote electronic voting on lower levels before going step by step towards more important elections until reaching the highest level (in Germany, the federal parliamentary elections). Thus, the technology should be tested in low level elections and can be improved if necessary; meanwhile the voters get used to the new technology and gain trust in it.

Electronic Voting as a Research Topic. The research community has investigated electronic voting since the early eighties. In 1981, one of the first technical research papers addressing electronic voting was published by David Chaum in [27]. Since then, numerous research papers have been published that propose cryptographic electronic voting approaches. Lists of publications in this area are available in [93], [90], and [59]. Blind signatures, homomorphic encryption, Mix networks, bit commitment, zero knowledge proofs, and threshold cryptography are only some examples of available cryptographic techniques. Most of them have been applied to solve the big challenge of providing unique voter identification (only eligible voters can cast a vote and those only once) and anonymous vote casting (the voter must be anonymous when he[1] casts a vote) at the same time. In later research papers, other aspects beside the pure voting protocol have been discussed, such as the trustworthiness of the voter's PC, Denial of Service attacks to the voting server, and temporary unlimited secrecy of the vote.

Besides this technical based research, electronic voting has also been analysed by other disciplines, including legal and social science. For instance, the time frame the secrecy of the vote must hold has been analysed from a legal point of view in [16], and a lot of research has been investigated on usability and accessibility issues of electronic voting systems (examples are [9] and [25]).

[1] Throughout this book the third person singular (him/he) is used as a gender neutral pronoun.

This research is as important as the technical literature because electronic voting is an interdisciplinary topic where people from different disciplines need to work together and learn from each other in order to successfully introduce electronic voting (see also [63]). This is reflected in many advisory boards for electronic voting projects where researchers from different scientific areas are represented. Also, the conference landscape has changed and conferences like the "Electronic Voting in Europe Workshop" in Bregenz were organised to hold an interdisciplinary and open discussion on all involved electronic voting issues.

Deployed Electronic Voting Systems. Beside the political, legal, and scientific developments, a huge variety of different electronic voting systems has been developed[2]. The most popular ones are the Diebold and Nedap electronic voting machines as well as the remote electronic voting systems POLYAS, VoteHere, Scytl, and T-Vote[3]. Such systems have been used for various elections all over the world on different political levels, both for trials and also for legally binding elections. Electronic voting machines have been used for parliamentary elections in countries like the U.S., the Netherlands, Belgium, France, Australia, Mexico, Brazil, Russia, and Germany. Remote electronic voting has been introduced in Switzerland, Estonia, the U.K., the Netherlands, Germany, France, and Austria, but on different political levels. In Estonia remote electronic voting is used for parliamentary elections countrywide, while in Germany it is implemented for elections in nationwide societies. A worldwide overview of countries where electronic voting machines as well as remote electronic voting systems have been applied is presented in [140].

In general, the systems in use are based on lower security techniques than the theoretical approaches published at various conferences. The deployed systems are easier to explain and to understand by the voter, while most of the theoretical available approaches require a rather technical and mathematical background. For instance, the Estonian electronic voting system can be explained easily as being the electronic copy of postal voting: The encryption of the vote corresponds to the inner envelope of postal voting, and the digital signature of the voter corresponds to the outer envelope (see [94] and Sect. 9.2 for more information). Thus, voters are rather convinced that this system works.

Activists. The introduction of electronic voting has not been embraced by everyone. Pressure groups[4] like Wij Vertrouwen Stemcomputers Niet[5] in the

[2] A list of electronic voting systems together with their links can be found in Sect. B.1.

[3] This paragraph might be a rather European or even German point of view.

[4] More information about these groups and their motivation can be found on their Web pages. Corresponding URLs are provided in Sect. B.2

[5] "Wij Vertrouwen Stemcomputers Niet" is Dutch and means "we do not trust voting computers".

Netherlands, the Chaos Computer Club (CCC) in Germany, the Verified Voting Foundation, and the Black Box Voting organisation in the U.S. as well as the group Irish Citizens for Trustworthy Evoting highly criticise electronic voting and are very sceptical with respect to the current electronic voting systems. They try to stop the usage of electronic voting in general and to prevent its introduction where politicians try to do so and they were even partly successful.

Responsible Election Authority. Despite the critical activist voices, "successful" hacking, and problems that have arisen, more and more responsible election authorities think about or have already decided to introduce electronic voting because of the manifold advantages it provides. Those advantages are, for instance, being more user-friendly, cost-effective and reliable, getting more accurate results and speed up the tallying time, as well as increasing the turnout by providing more mobility to the voter on election day. The responsible election authority naively believes that the insecure electronic voting systems are only those from other responsible election authorities or those from other countries. Thus, they argue that though the system might not be perfectly secure, first, the effort to hack the electronic voting system is too high compared to the importance of the election (for example, elections in societies), and, second, the paper-based system is also not one hundred percent secure. From their point of view electronic voting is acceptably secure without understanding all technical details. This problem has been addressed in literature (for instance in [102]), where the authors try to help the responsible election authority to "better understand the perspectives of electronic voting sceptics [..] to help them understand what the electronic voting sceptics are saying and why they are saying it, and to appreciate some of the questions about electronic voting technologies that worry many technologists".

1.2 Motivation

History shows that electronic voting cannot be stopped in our technically oriented society, where an increasing number of processes are mapped into the electronic world and voters become more and more mobile. Whenever people and in particular the responsible election authorities see the various advantages, they will start to implement electronic voting. It is only a question of time till electronic voting will be used for more and more elections and voters will become more aware of it. Maybe our grandchildren will not believe that we were used to use pen and paper to cast a vote and to go to a polling station (or even will not know anymore what a polling station is). Thus, it is essential to investigate how we can provide a *trustworthy base for secure electronic voting* in order to protect our democracy in the future and avoid the application of insecure electronic voting systems. Accidental as well as malicious abuse of electronic voting in our future election form must be prevented.

Considering the fact that the introduction of electronic voting has already started and a lot of problems with existing electronic voting systems have already been detected, the important questions are which of these existing *electronic voting systems are secure (enough)* to use them for a particular election and, related, how to design an electronic voting system that is secure (enough) for the application in certain elections. To answer these questions, the following two tasks are essential:

- First, the definition of the term “secure (enough)” for a particular election: as always in security, there is no 100% secure electronic voting system. Therefore, it needs to be defined whether a system is secure (enough) for a specific election in a certain environment; that is, under a certain *trust model*. Thus, a standardised, consistent, and exhaustive list of requirements for the electronic voting system which serves as basis for any evaluation, needs to be defined.

  Definition (Trust Model): In this book the definition of a trust model is divided into the following three aspects:
  - What are the assumptions about the environment?
  - What are the intruder's technical capabilities?
  - Who can be trusted not to maliciously cooperate with others?

  Note, a system that ensures the requirements regarding a minimised trust model, provides the most security, as the minimised trust model contains the lowest amount of assumptions and the maximised intruder's technical capability. Vise versa, the maximised trust model requires the highest amount of confidence in the environment and the lowest intruder's capability.
- Second, the assurance that a particular electronic voting system ensures these requirements under the defined trust model. Thus, a standardised *evaluation methodology*, which guides the evaluator how to check a given electronic voting system for a particular trust model. Elections are of individual importance, therefore it is essential to use an evaluation methodology which supports different evaluation depths and thus different assurance levels.

Having such a framework, the responsible election authority would need to define the underlying trust model and the assurance level. Based on this framework, a professional evaluator could examine a certain electronic voting system and decide whether it can be used in this context because it is secure enough or not.

1.3 Contribution, Methodology, and Structure

The contribution of this book is the development of such a framework for the evaluation of electronic voting systems. To reach this goal, four major

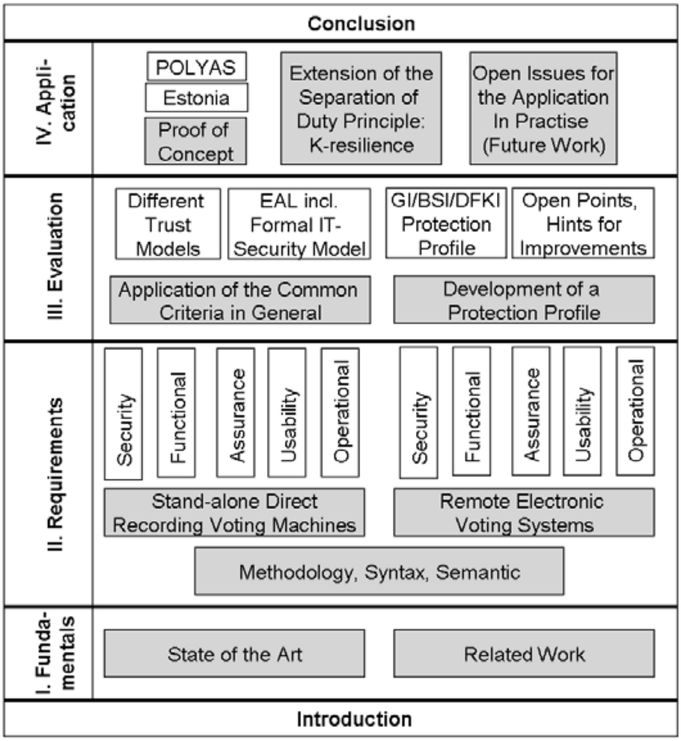

Conclusion
IV. Application
POLYAS
Estonia
Proof of Concept
Extension of the Separation of Duty Principle: K-resilience
Open Issues for the Application In Practise (Future Work)
III. Evaluation
Different Trust Models
EAL incl. Formal IT-Security Model
GI/BSI/DFKI Protection Profile
Open Points, Hints for Improvements
Application of the Common Criteria in General
Development of a Protection Profile
II. Requirements
Security
Functional
Assurance
Usability
Operational
Security
Functional
Assurance
Usability
Operational
Stand-alone Direct Recording Voting Machines
Remote Electronic Voting Systems
Methodology, Syntax, Semantic
I. Fundamentals
State of the Art
Related Work
Introduction

Fig. 1.1. Structure of the book

steps are taken, which correspond to the four main parts of this book (starting after this introduction chapter): Fundamentals, Requirements, Evaluation, and Application. Followed by these four parts, there is a conclusion part and an appendix. The contribution of each part, the applied methodology, and its structure are described in this section. Furthermore, the structure of the whole book is illustrated in Fig. 1.1.

- The **first part "Fundamentals"** provides the reader with a general introduction to the relevant issues of electronic voting and a critical discussion on available requirements and evaluation documents for electronic voting systems. Accordingly, this part is divided into the following two chapters:
  (A) In Chap. 2, the state of the art in electronic voting is proposed, analysed, and structurally reworked. A classification of election forms (including paper-based and electronic voting) and general differences between the latter two voting systems are worked out. Moreover, different implementations of the most common electronic voting machines in use, as well as remote electronic voting systems are described and discussed.

(B) Chapter 3 discusses and analyses related work in terms of a couple of available lists of requirements and evaluation methodologies for electronic voting systems, which are conducted in so called requirement catalogues, corresponding laws, or regulations. Furthermore, scientific work concerning requirements for electronic voting is included in this discussion. The contribution is an overview of existing requirement documents, including the proposed evaluation methodology, and a list of vulnerabilities, which need to be overcome in the framework proposed in this book. In addition, some of the available requirement documents are identified as basis for the requirement definition in the second part of this book.

- The **second part "Requirements"** contributes a standardised, consistent, and exhaustive list of requirements for electronic voting systems. This list is detailed enough for the system developers to understand how their system can meet these requirements. In addition, the requirement list is deduced from legal input, so that lawyers accept these requirements. The provided list of requirements is not considered to be just another requirement catalogue but it is developed to improve and combine existing literature (which is referred to for each requirement). Therefore, a particular syntax and semantic is proposed and a particular methodology to develop the requirements is applied. This second part addresses two forms of electronic voting: stand-alone direct recording electronic voting machines and remote electronic voting systems. Correspondingly, this part is divided into the following three chapters:

  (A) Chapter 4 first describes the applied methodology to develop the standardised, consistent, and exhaustive list of requirements. The methodology includes cross checks against existing catalogues, the election principles, and possible threats. Furthermore, the necessity of defining different lists for different categories of electronic voting is explained. In addition, the language and notation for the requirement specification is defined in this chapter: a set of definitions for voting terminology and in particular for electronic voting specific items is presented. Moreover, the applied syntax and semantics are defined. This clarification enables a unique application of (electronic) voting specific items and a standardised language to ensure accuracy and contingency and, thus, to facilitate comparability.

  (B) The fist result of the development process – a standardised, consistent, and exhaustive list of requirements for stand-alone direct recording electronic voting machines – is provided in Chap. 5. This list contains system requirements (divided into functional, security, and usability requirements), organisational requirements, and assurance requirements.

  (C) The second standardised, consistent, and exhaustive list of requirements for remote electronic voting systems is provided in Chap. 6. As described for Chap. 5, this list contains system requirements (divided into functional, security, and usability requirements), organisational requirements, and assurance requirements.

- The **third part "Evaluation"** contributes the proposal and discussion of a standardised evaluation methodology and certification procedure. This methodology accounts for the above defined security, functional, and assurance requirements, while it does not address operational and usability requirements. A further constraint concerns the type of electronic voting system: the evaluation methodology is elaborated for remote electronic voting systems. However, having defined a corresponding evaluation methodology, it can easily be adapted and/or extended for electronic voting machines or other forms of electronic voting.
  The developed evaluation and certification methodology provides a systematic and well defined compliance check for the selected requirements. In addition, the methodology provides impartial, comparable, and repeatable evaluation results, and is flexible with respect to the evaluation depth and the underlying trust model. Thus, arbitrary types of elections with their different trust models and different requirements to the evaluation depth can be handled. As it is not possible to handle all these options in one framework, a common basis framework for all remote electronic voting system evaluations is described. Accordingly, this part is divided into the following tow chapters:
  (A) Chapter 7 first analyses different existing IT security evaluation standards and shows that the Common Criteria [35] in combination with the Common Evaluation Methodology [36] works best for the evaluation and certification of remote electronic voting systems according to the defined security, functional, and assurance requirements. Second, this chapter applies the Common Criteria to remote electronic voting. This is done in the following four steps:
    - First, the Common Criteria itself is explained in detail and a mapping between the syntax used in the first part of the book and the Common Criteria language is provided.
    - In the next step, the role of a trust model for remote electronic voting in the context of the Common Criteria is discussed. A general as well as a detailed analysis id done for two essential remote electronic voting examples: the 'temporary unlimited secrecy of the vote' and the 'trustworthiness of the vote-casting device'. In this part, different possibilities to define the trust model are presented and the consequences for each possibility are presented.
    - The third step focuses on the evaluation depth. The previously identified assurance requirements for remote electronic voting systems are translated into the Common Criteria language.
    - The last step focuses on achieving high assurance levels: In this context, the Common Criteria requires a formal system specification and in particular a formal IT security model against which the system is evaluated. This is a first step to develop such a

formal IT security model for remote electronic voting systems. It covers an initial set of security and functional requirements. Note, formalising the previously defined security and functional requirements in a formal IT security model has more advantages than enabling corresponding Common Criteria evaluation depths: with the development of such a model the presented list of security and functional requirements can be validated or if necessary further improved by such a fundamental consistency check.

(B) Chapter 8 focuses on the completion of the evaluation framework. According to the Common Criteria, the security, functional, and assurance requirements are composed to a Protection Profile (a Common Criteria specific implementation-independent statements of security needs), however not a universal Protection Profile is provided but a core Protection Profile[6]. It is based on the lowest acceptable evaluation depth and the maximised trust model. This core Protection Profile should/has to be satisfied by any remote electronic voting system but can be extended if the trust model and/or the required evaluation depth change.

- The **fourth part "Application"** contains the application of two available remote electronic voting systems to the core Protection Profile and deduces open points within the application of the provided evaluation framework: it is shown that the third aspect of the trust model 'who can be trusted not to maliciously cooperate with others' is not sufficiently addressed by the core Protection Profile. Thi fourth part also proposes the remaining points for an application in practise. Correspondingly, the fourth part is *divided* into the following three chapters:

  (A) In Chap. 9, a proof of concept is executed to show the validity of the framework. Here, two remote electronic voting systems are analysed with respect to the previously developed core Protection Profile: the POLYAS system from Micromata and the Estonian remote electronic voting system, which was used for the parliamentary election in March 2007.

  (B) Chapter 10 concentrates on the separation of duty principle, which is neglected by the Protection Profile as it aims to be generic for the application of any voting protocol. An additional mechanism is developed to calculate how many entities need to maliciously cooperate in order to violate a particular security requirement: the calculation of the k-resilience value is introduced and is recommended as an extension to the Common Criteria certificate.

  (C) Chapter 11 addresses open issues for an evaluation of electronic voting systems in terms of future work.

[6] This core Protection Profile is based on the GI/BSI/DFKI Protection Profile [161].

The **conclusion** part closes the book with a summary of the contribution and implications for the trust in electronic voting systems. The **appendix** contains links to Web pages from electronic voting system vendors and electronic voting antagonists, the glossary, those requirements that are not considered, and the structure of a Protection Profile.

Part I

Fundamentals

2

Implementations of Electronic Voting

In order to define requirements for electronic voting systems it is necessary to understand electronic voting in general and in particular its possible implementations. Thus, this chapter provides an overview and classification of existing (paper-based and electronic) election forms, including multiple channel voting. The second part of this chapter presents the main differences between paper-based elections and electronic voting. The next two parts analysis in detail those forms of electronic voting relevant for further discussions in this book: Thus, the third part of this chapter proposes and discusses electronic voting machines (direct recording electronic voting machines and the Digital Election Pen). The fourth part focuses on remote electronic voting systems (including possible implementations of voter authentication, the secrecy of the vote, and the client-side voting software).

2.1 Classification of Election Forms

This section introduces in Sect. 2.1.1 three dimensions to categories paper and electronic election forms. Sect. 2.1.2 proposes existing forms and categorises them according to Sect. 2.1.1. Multiple channel elections (meaning the combination of different election forms) are discussed in Sect. 2.1.3.

2.1.1 Dimensions

Three fundamental dimensions to categories election forms are medium, environment, and point in time (according to [150] and [151]). In the following, these dimensions including possible implementations are proposed:

- The *medium* used to hold the ballot:
  - The traditional medium *paper*, denoted by ‘p’ (in this sections).
  - The new *electronic* medium where votes are electronically stored (in bits and bytes), denoted by ‘e’.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 13–35, 2009.

  Definition (electronic voting): Electronic voting covers any election form where at least at one point in time an electronic copy of the vote is stored electronically and the election result is computed based on the stored e-votes.
  - The *mechanical* medium where votes are recorded by mechanical components.

    Representatives are mechanical devices like lever machines which are based on patents by Thomas Alva Edison. There is no computer interface for electronic tabulation but analogously to odometers the number of votes for a particular candidate is incremented each time a voter moves the lever that correspond with the particular candidate. At the end of the election, the poll workers open up the back of each device to read the counting wheels and determine how many votes were cast for each candidate. These mechanical voting devices are the precursor of electronic voting systems and many of them have been replaced nowadays by electronic voting machines. Therefore, mechanical voting devices are not further discussed in this book (for more information see, for example, [40, 75] or section 2.1 in [101]).

- The *environment* where people cast their vote:
  - In a *controlled* environment where poll workers ensure the accuracy, privacy, and integrity of the vote casting process. This category is also called presence voting and is denoted by ‘c’. Controlled environments are for instance, traditional polling stations, post offices, or embassies for people living abroad.
  - In an *uncontrolled* environment where no officials like poll workers ensure the accuracy, privacy, and integrity of the vote casting process but the voter himself has to do so. This category is also called absentee voting and is denoted by ‘u’. Casting a vote in an uncontrolled environment (for instance, at home) means that the voter has no direct contact with election officials. Thus, it provides an opportunity for family voting (family members influencing each other or filling out the absentee ballot of other family members), voter coercion, and vote buying (selling the blank absentee ballot).
- The *point in time* when vote casting is enabled:
  - On *election day*, denoted by ‘ED’;
  - *Prior to election day* (also called early voting or voting in advance), denoted by ‘A’.

  Some voting forms can be mapped to both categories of the dimension point in time depending on the implementation (for instance, postal voting in Austria).

2.1.2 Categories of Election Forms

At least the following eight different election forms can be identified:

Traditional Polling Station Election [p, c, ED/A]

In a traditional polling station election voters cast their votes from official polling stations on *election day*. Polling stations are in general housed in public buildings, such as schools. Each voter is assigned to a particular polling station. In the polling station, poll workers verify the voter's right to vote and hand out a *paper ballot*. The voter makes his choice in a polling booth and, thus, in a *controlled* environment. The voter puts his vote in a ballot box and leaves the polling station.

Besides the popular implementation on the election day, there exists three further exceptions where voters can cast their vote in *advance*: in some countries, voters living abroad have the possibility to cast their vote in an embassy (for example, voters from France, Germany, and the Netherlands). There are other countries which allow voting in advance in particular post offices (for instance, Sweden) or in particular polling stations (for instance, Estonia).

Home Voting [p, c, ED/A]

Home voting enables sick people, who are not able to come to the polling station, to cast their vote remotely (implemented, for example, in Estonia). Here, some of the poll workers visit these people on *election day* or *in advance*, give them their *paper* ballot and ensure the possibility to make their choice in a free and secret environment, and thereby providing the *controlled environment*[1] at home. The vote is put in a special ballot box and later mixed with the votes cast in the polling station.

Postal Voting [p, u, ED/A]

Using postal voting, the voter makes his choice on a *paper ballot* at any location he wants. There are *no poll workers ensuring* that the voter can cast his vote unobserved, without any influence and without any coercion; this is up to the voter. Due to the time and the way the election envelope needs to arrive at the central ballot box, postal voters needs to cast their vote in *advance* if the postal votes need to be calculated on election day (for instance, in Germany). In other countries like Austria, postal votes can also be sent on the *election day* because the law demands to wait several days after election day before tallying postal votes.

Postal voting was introduced to enable more people to take part in the election, like sick and old people as well as those traveling or living abroad on election day. Thus, the principle controlling election laws demanding a universal franchise has more priority here than freedom of voting and secret elections. To balance these concerns, in many cases voters need to request postal voting and it is only implemented as exception (like in Germany).

[1] Some might argue that depending on the trustworthiness of the poll workers the category 'controlled environment' should be substituted by the category 'uncontrolled environment'. However the reason why a country introduces home voting is because they trust their poll workers.

Paper-Based Electronic Voting Systems [e, c/u, ED/A]

In paper-based[2] electronic voting, voters cast their vote on paper, but proceeding (meaning scanning, recording and storing as an *e-vote*) and counting of votes happens electronically. Scanning can be done directly after vote casting or at the end of election day. The vote is either cast in a *controlled* or in an *uncontrolled* environment; either on *election day* or in *advance*.

The advantage of such systems is that in the case of technical problems the paper votes can still be counted by hand and a re-election would not be necessary. The disadvantage is that the responsible election authority must decide and define in the election regulations whether the paper votes or the e-votes are the proper votes. In the case where the paper ones are the legal one, the fast tallying of the e-votes only provides unofficial results and can be compared to an extrapolation. It needs to be subsequently confirmed by tallying the paper votes. The case of a difference between the e-result and the paper result is problematic. Here, the error rates of the reader need to be taken into account. The most popular classes of paper-based, electronic voting systems are:

(1) Punch card voting systems. Voters punch holes in the ballot (with a supplied punch device) opposite their choice. Afterwards, the vote is read into the tabulating device and the voter places the paper-vote in a ballot box. Here, in addition to the error rate of the reader the sloppiness in completing the punch needs to be considered, too. For more information see, for example, [75, 130, 169], and [25]. Example systems are: Portapunch machines, Datavote, and Votomatic systems.

(2) Optical scan systems (also called marksense systems). Optical mark-sense scanners were developed to administer college entrance exams and other standardised tests and were later applied to elections. In such systems, the voter casts his vote on paper usually by filling a rectangle, circle or oval, or by completing an arrow. These paper votes are read by an optical scanner. The optical mark-sense ballots appear very similar to the classical ones. In addition to scan errors, optical scan systems are challenged by wrongfully marked paper forms. Marks not according to the rules might cause wrong vote counts or other errors. For more general information see, for example, [75]. The Digital Election Pen system is the newest example for this category and is further discussed in Sect. 2.3.2. Other representatives are the Prêt à voter system (see [31] and [127]), the Three-Ballot system (see [125]) and the Scantegrety System (see [132]). All these system are based on academic work aiming to implement voter and universal verifiability (see also Sect. 4.5). However, the systems have also disadvantages: for instance, the candidate order in the Prêt à voter system needs to be randomised for each ballot. Moreover, the voter

[2] In other classifications, like in [23], these election forms are designated as a paper medium. But according to the definition for electronic voting is this book, paper-based electronic voting systems need to be classified as 'electronic'.

needs to discard the left-hand column carrying the candidates and to keep the right-hand column with his cross as receipt. He has to use this piece of paper to later verify his vote. Such additional procedures are hard to explain to voters and might confuse them.

Stand-Alone Electronic Voting Machines in Polling Stations [e, c, ED/A]

Stand-alone electronic voting machines are installed in polling stations. These are *electronic* devices to cast votes, often by using a touch screen[3], and to store votes locally. These devices are terminals similar to bank cash machines, which are located in polling booths and, thus, in a *controlled environment*. Moreover, they are not connected to the Internet or any other network. The authorisation of a person as an eligible voter is still done on paper like in traditional polling station elections. Analogously to traditional polling station elections, the stand-alone electronic voting machines can be used for voting on *election day* as well as in *advance* in embassies, post offices or specific polling stations.

Networked Electronic Voting Machines in Polling Stations [e, c, ED/A]

Networked electronic voting machines in polling stations are electronic voting machines connected to an network like the internet. This enables voters to cast their vote in an arbitrary polling station. Thus, this election form provides more flexibility to the voter than other polling station election forms. The voter's right to vote is checked online against a central electoral register. This can be done either without an electronic authentication token but still by poll workers based on traditional mechanisms, or by the networked electronic voting machine[4].

Kiosk Electronic Voting Machines [e, u, ED/A]

Kiosk electronic voting machines are similar to networked electronic voting machines in polling stations. The difference is the location: the kiosk version is not arranged in a polling station but at central places like schools, universities, libraries or super markets. Thus, the first step to an uncontrolled environment is made as poll workers do no longer ensure the accuracy, privacy, and integrity of the vote casting process. Due to the absence of the poll workers, the machines need to be protected like bank cash machines.

[3] The Nedap machines are a popular example for stand-alone electronic voting machines working without a touch screen.

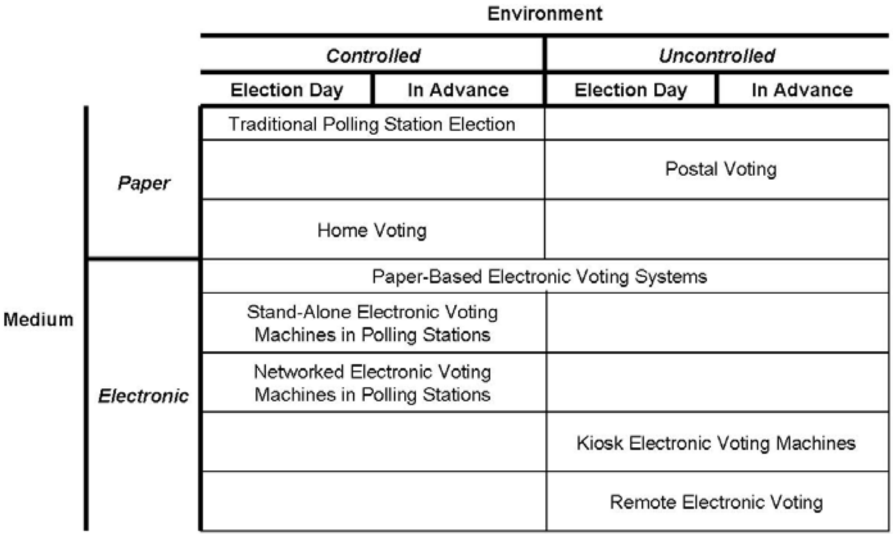
[4] Systems where only a voter's right to vote is checked online but the vote is cast on paper, do not belong to electronic voting systems. In the categorisation of this book such election forms are classified as a special case of traditional polling station elections.

Remote Electronic Voting [e, u, ED/A]

Remote electronic voting is the *electronic* counterpart to postal voting. In some literature it is also called Internet-Voting or Online-Voting (especially in German: "Online-Wahlen"). Here, the voter can choose any place to cast his vote (*uncontrolled environment*) as long as he has a device connected to the network. Such a device could be a PC, PDA or mobile phone[5]. Remote electronic voting can only be applied if the authorisation check as well as the vote casting is done online and, thus, electronically. Thus, it is the most complex and difficult election form as it poses the problems of uncontrolled environments combined with all of the problems of direct recording electronic (DRE) voting machines. In addition, the client-side voting software runs on a machine and an operating system outside the control of the responsible election authority. An advantage compared to postal voting is the short transportation time. It is possible to implement remote electronic voting to either cast a vote in *advance* or on *election day* and still tally the e-votes on election day.

Overview

Figure 2.1 provides an overview of the above described election forms and their classification according to the dimensions from Sect. 2.1.1.



		Environment			
		Controlled		Uncontrolled	
		Election Day	In Advance	Election Day	In Advance
Medium	Paper	Traditional Polling Station Election			
				Postal Voting	
		Home Voting			
	Electronic	Paper-Based Electronic Voting Systems			
		Stand-Alone Electronic Voting Machines in Polling Stations			
		Networked Electronic Voting Machines in Polling Stations			
				Kiosk Electronic Voting Machines	
				Remote Electronic Voting	

Fig. 2.1. Election categories (categorised according to the three dimensions)

[5] Voting using mobile phones and the short message service (SMS) is also a possible approach, which has been discussed especially in the U.K. trials. The application of interactive TVs is sometimes added to this category but interactive TV and SMS voting are not discussed in this book.

2.1.3 Multiple Channel Elections

In today's traditional elections, in most cases at least two different forms are applied. For example, Germany has implemented for federal elections traditional polling station election on election day and in advance, as well as postal voting. Such elections, enabling two or more different election forms at the same time are called multiple channel elections.

Multiple channels were introduced to ensure a universal election. However, implementing multiple channels produces organisational overhead because multiple vote casting using different channels needs to be prevented. Depending on the system this can become arbitrary difficult and time intensive. For instance, in Sweden, voters are allowed to cast a vote in advance and replace this vote by casting a vote on election day. Here, it takes days to decide for each advance vote whether it can be counted or it was superseded by a vote cast on election day.

These procedures become even more complicated with the introduction of remote electronic voting as this election form may, in accordance with current electoral laws, not be applied as the only voting channel[6], even for elections where currently only postal-voting is applied (for instance elections of the governing boards of social security institutions in Germany). The reason is that it cannot be assumed that each voter has access to the remote electronic voting system or is able to use it (see [168] and [79] for a legal discussion on this issue). Taking a look at the Estonian implementation of multiple channels, including remote electronic voting (for detailed information see [95] and [106]), the complexity becomes obvious: the voter could cast an e-vote and update this vote by using the remote electronic voting (more about vote updating in Sect. 4.5) or by going to a traditional polling station. The electronic form of vote updating can be handled automatically by the system. The e-votes of those who also cast a paper vote must be manually deleted in the system. For a more detailed discussion of the security issues of multiple channel elections see [170].

2.2 Paper-Based Elections versus Electronic Voting

As the dimension "medium" is the dimension which is most important for this book, general differences between paper-based elections and electronic voting are presented in this section:

Paper-based systems are easier to manipulate than electronic voting systems. An attacker does not need technical knowhow or any other expertise. Vise versa, electronic voting systems are very complex and complicated. Thus,

[6] There is one exception known: the elections in the 'Initiative D21' (the Initiative D21 is Germany's largest public private partnership). But here all voters needed to sign in advance that they have got a PC which is connected to the Internet and that they are willing to use the remote electronic voting system (for more information see [81]).

without an established technical and cryptographic background, an attacker has no chance to manipulate an electronic voting system. The degree of damage an attacker can inflict, is exactly the reverse case: while within paper-based systems only small alterations to the election result are possible, an attack against an electronic voting system and in particular a remote electronic voting system can manipulate the entire election result. In paper-based systems one person could, for example, add or delete votes in one polling station (if the poll workers agree by unfair means) or delete postal votes in one district. In electronic voting systems, manipulated devices can be deployed or a Trojan Horse can be distributed to a remote electronic voting system. Such attacks could cause that the content of a vote is altered before casting it.

Another identified difference is the transparency and voter's trust in the system: The average voter understands paper-based systems (for instance, the necessity of the ballot box and the need for two envelopes in postal voting). To maximise transparency in paper-based elections, some countries like Germany (see [22] for the corresponding legal article) allow voters to observe the entire procedure in the polling stations including the check that the ballot box is empty in the morning, that no manipulations happen during the day, and that the tabulation is correct. Even if a voter does not observe the whole process himself, he trusts the procedure because the present poll workers have either different interests in the election result or are neutral and, thus, control each other. With electronic voting systems there is no real observation possible by the voter. Observing the tallying within an electronic voting system is like pressing a button of a black box. This weakens the transparency. Moreover, trust in poll workers does not really help because they play only a supervisory role, since they cannot actually observe the internal functioning of the electronic voting machine or the voting server.

The next identified difference is the list of people involved, that the voter needs to trust. While in paper-based systems, there are only poll workers (including the talliers), electronic voting systems need the following additional groups: the software developers, people who host and administrate the voting servers and the server-side voting software.

The literature also identifies re-tallying as a difference between electronic voting and paper-based elections. Re-tallying is done once in the polling stations to ensure that miscounts can be detected. The responsible election authority can order a recount, for instance, in the case of a close run or refutations. A recount of paper votes will arrive at a slightly different result, due to human errors. With electronic voting re-tallying becomes much less powerful: a recount of the same vote storage with the same tallying software ends up with the same result. Even using a second, redundant vote storage ends up with the same result, as this storage is not independently recorded, but created by the same voting software which created the original records. Exceptions to this are paper-based, electronic voting systems. Here, a manual recount is a reasonable way to check the correctness of the system.

A last identified category of differences between paper-based and electronic voting concerns the voter interface: the presentation of the ballot is different between traditional paper-based election systems and electronic voting systems. In paper-based election systems, the ballot looks the same for all voters while in remote electronic voting systems the electronic ballot interface differs from vote-casting device to vote-casting device. Obviously, a ballot cannot be equally displayed on a notebook as on a mobile phone. Large ballots, as used in some districts of Germany (for instance, in Hamburg) or in Belgium, cannot be displayed at once on the screen. Another interface difference is given by the intelligence of electronic systems. The electronic voting system can, for example, warn the voter if he inadvertently tries to spoil his vote because he made too many or not enough choices. Related to this difference is the identification of invalid votes. While there are cases in the paper-based system where poll workers have long discussions about the vote‘s validity, in electronic voting systems, each vote is unambiguously valid or invalid.

The main identified differences are summarised in Table 2.1. A detailed comparison is provided in [101]. In [86] and [87] postal voting and remote electronic voting are compared.

Table 2.1. Differences between paper-based elections and electronic voting

Paper-based elections	**Electronic voting**
Easier to manipulate	Technical know-how necessary
Decentralised	Centralised
Only small alterations to the result	Alterations to the entire result
Very transparent and observable	(Evaluated) black box
Trust in poll workers	Trust in poll workers, system developers, adminis
Meaningful re-tallying	Electronic re-tallying based in same input
Equal ballot sheets	Different ballot layouts (depending on the device)
Sometimes hard to decide about the voter’s will	Unambiguously valid or invalid votes

2.3 Examples of Electronic Voting Machines

This section proposes and discusses two poüular representatives of electronic voting machines: direct recording electronic voting machines and the Digital Election Pen system.

2.3.1 Direct Recording Electronic Voting Machines

Direct recording electronic (DRE) voting machines can either belong to standalone electronic voting machines in polling stations, to networked electronic voting machines in polling stations, or to kiosk electronic voting machines. In this book, only those DRE machines used as stand-alone electronic voting machines are considered; that means votes are cast at a dedicated voting device which is not networked. Any data from these devices can only be transferred to other devices by physically transporting some storage medium.

According to [131], stand-alone DRE machines were introduced by Shoup and Microvote. The first representatives emulated classical lever machines to the extent that unfolding the electronic voting machines created a complete polling booth. The levers and electromechanical counters were replaced by push buttons and micro-processor software. Nowadays, flat panel displays with mechanical or electro-optical components (typically buttons or a touch screen) are provided to choose a candidate and cast an e-vote.

No receipt or confirmation is given to the voter to later verify the correct tally of his vote. Thus, the voter needs to trust that the desired vote has been entered correctly into the memory while an independent proof like a physical record is not provided to the voter. Hence, the voter needs to trust the programmers who developed the electronic voting machine and the poll workers that these machines have not been manipulated after the deployment.

A DRE system processes e-votes by means of a computer program. The e-votes are stored locally in memory components, as is ballot information. The tallying software can either be a part of the DRE system or run on another device. Usually, after finishing the tallying, a copy of the election result is printed for the poll workers.

For more information see, for example, [40] and section 2 in [101]. The best known representatives of digital recording electronic voting machines are developed by Diebold, ES&S, and Nedap.

2.3.2 Digital Election Pen

The Digital Election Pen is the newest approach to implement paper-based electronic voting machines. The state government of the Free and Hanseatic City of Hamburg, Germany, was planning to introduce this new type of electronic voting machine for its Bürgerschaft (state parliament) election in February 2008. Shortly before the election, the state parliament decided not to use the pen solution due to negative press and security reservations.

The idea is to use a digital pen to mark the paper ballot in the same way as a common pen. The digital pen is slightly larger, because of an integrated camera besides the usual lead. Using a digital pen for elections does not introduce essential changes to the voting procedures and its handling does not make a big difference for the voter: he still marks his choice on a paper ballot in the polling booth and the pen stores the corresponding positions. At the end of the individual voting process the voter drops the paper ballot into the ballot box as usual and additionally inserts the digital pen into a docking station. The voting data from the pen is copied(in a randomised order) to an electronic ballot box and erased from the pen. Afterwards, the pen is re-initialised for the next voter.

The scanning of the voter´s mark is done based on a thin pattern (railing 0.3 mm) on the paper ballot. The contact of the pen on the paper is noticed by a sensor in the top of the pen. This causes the integrated camera to start scanning. Using the pattern, the pen deduces its current position and stores it. This technique was developed by the Swedish company Anoto Group AB.

The election result tallying is based on an interpretation of each ballot. Rules for separating valid from invalid votes have to be specified. Based on this, the software distinguishes three cases: valid votes, invalid votes, and those votes that have to be checked manually by the poll workers because they are not clearly marked.

In the Hamburg case it was planned to use the digital pen in the following way: E-votes are stored on a specially prepared notebook. Three docking stations are connected to the notebook; one to initiate the pen, one to store the vote and delete the pen, and one to cancel the vote and to delete the content on the pen. A memory stick is used to provide redundant storage and an ordinary printer is in place to print out status reports and the final result.

One of the main matter of dispute was the question which vote is the legal one: the electronic or the paper vote. The project was canceled before making a decision on this issue. More information about the Digital Election Pen and the applied evaluation techniques can be found in [159], [158], and [7].

2.4 Overview of Remote Electronic Voting

On a high level view the architecture for almost all of the existing remote electronic voting systems and proposed voting protocols is similar and illustrated in Fig. 2.2: the involved parties are voters, administrators, and the poll workers. The encompassed components are:

- The vote-casting device, running the client-side voting software. The vote-casting device can be any electronic device connected to the network, a PC, a notebook, a PDA or a mobile phone.
- The voting server running the server-side voting software. Most of the remote electronic voting systems distinguish between several different voting servers. For instance, voting servers responsible to check a voter's right to vote in the electoral register (the so called registration servers - RegServer) or the ballot box servers[7] (BBServer), which store the e-votes in their e-ballot boxes (EBB).
- The tallying software that is running on a separate component that is not connected to any network.

[7] In some remote electronic voting systems the ballot box server is called bulletin board because it has special properties. The bulletin board provides a designated field for each voter, where he can write to (for instance, authorised by signed messages from the voter). No messages can be deleted from a bulletin board and everyone has read-access to the whole board.

The communication between these components runs over a network, for instance, the Internet or a particular intranet.

Most of the existing remote electronic voting systems and proposed voting protocols are covered with a general architecture presented in Fig. 2.2. However, there are two exceptions which work without a central voting server:

- Approaches sending messages in several rounds from voter to voter (see, for example, [43] and [3]). These approaches can only be used within very small voter groups because the amount of messages to be sent grows exponential in the number of voters.
- Approaches based on architectures that use a peer to peer (P2P) Web cache to enable the voter to cast a vote and to deposit it in the P2P file sharing network for collection by the responsible election authority (see [24] for detailed descriptions).

These two approaches are not further taken into account in this book.

Three Dimensions of Remote Electronic Voting. Remote electronic voting can be classified along the following dimensions: the authentication technique used for unambiguous identification and authentication of the voter (see Sect. 2.4.1), the way the secrecy of the vote is ensured (see Sect. 2.4.2), and the used vote-casting device (see Sect. 2.4.3). These dimensions are presented and discussed in the corresponding sections.

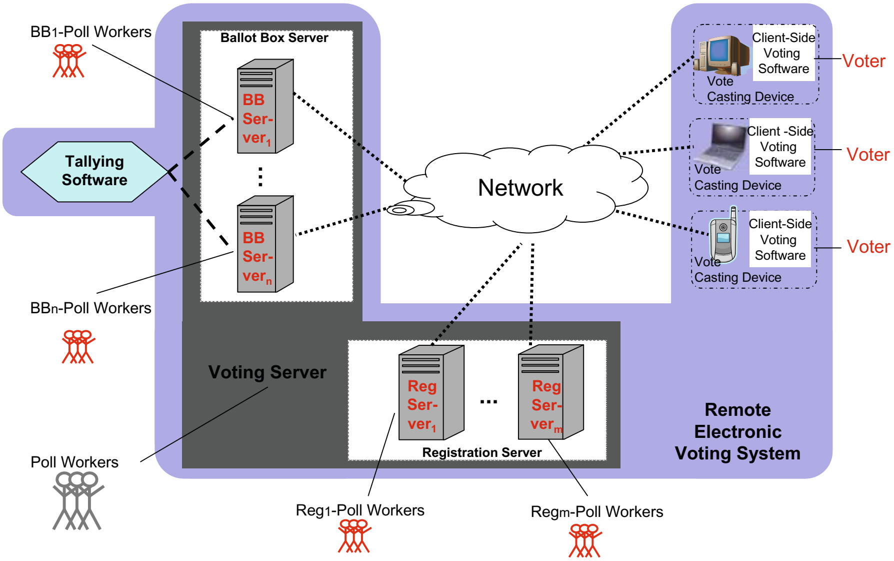


Fig. 2.2. General high level architecture of a remote electronic voting system

2.4.1 Authentication Techniques

Every remote electronic voting system needs to implement voter identification and authentication techniques to ensure that only eligible voters may cast a vote and those only once. In information security, mainly three ways of identification and authentication are known (as well as corresponding mixed ones): something you know, something you have, and something you are. All three techniques are analysed in the following paragraphs with respect to their allocatability for remote electronic voting (taking security, usability, and economic aspects into account).

Something You Know: a Secret

The first category is based on knowledge, while two different implementations are possible:

- The first possible implementation of voter identification and authentication is applied in accordance with the set up of an e-mail account: in the election setup phase, it is possible to set up a voter account, which will be later used by the voter to cast his vote. Although this approach may be easy from the voter's perspective, it has three weak points: first of all, it cannot be excluded, that other persons, who are not authorised for this particular election, set up an account. Second, voters might choose weak passwords which can be easily hacked by an intruder. Third, vote buying cannot be excluded, because voters could easily send electronically their login data to a potential buyer.
- A further type of identification and authentication through knowledge of a secret is called vote-TAN procedure. The vote-TAN, a per voter unique code of letters and digits, is send by post[8] to eligible voters in the election setup phase. This variation is rather similar to the above one with respect to the usability issues. However, the costs increase since the eligible voters get their TAN by post. But the security increases because only eligible voters have a TAN and this can be generated through the responsible election authority as a strong "TAN". The risk that the TAN will be handed on to an intruder in order to sell the vote still exists.

Something You Have: a Token

The second category is based on ownership, while two different implementation possibilities can be identified:

- In the one implementation, a new election specific identification and authentication card is used, which will be send to the voter prior to the election (similar to the TAN from the above category). Compared to the TAN solution this one provides more security since the buying of votes is more expensive because getting the card is more difficult than getting a

[8] It is assumed that the post channel is secure.

copy of the TAN, for example, via email. The costs rise substantially: apart from production and distribution cost of the cards, substantial costs of an appropriate card reader arise. From the usability point of view, negative affects appear caused by the necessary installation of the card reader and corresponding software on the voters PC.

- In the second implementation, a pre-existing election identification and authentication card is used, which the voter already owns and uses for identification purpose in other areas, like his ID card, job card or library card. Some of the above disadvantages can be eliminated by the employment of a pre- existing card. Such a card will not be lightly passed on to a vote buyer, since this automatically means that all other applications of this card are passed on as well. Additionally, the use of an already owned card increases the user-friendliness. However, the costs of the card reader remain, if the voter does not possess such a device, yet.

Something You Are: Biometrics

The third authentication category is based on biometric attributes (their applicability to electronic voting is discussed in detail in [68]). Examples of biometric attributes are finger prints, iris scans, face recognition (size and position of different facial features), voice (mode and tone while speaking), manual signature (form and dynamic aspects), and DNA. The form or structure of each of these attributes is unique for one person. In order to authenticate a person the corresponding attribute is scanned. The scanned copy of the attribute is then compared to the one stored of this person. In case it matches, the person is authorised otherwise he is rejected. The major point of concern for a biometric system is how to securely store such sensitive data.

The main advantage of biometric authentication is that attributes cannot be forwarded to another person, for instance, vote buyers. Unfortunately, the matching of scanned and stored data does not work perfectly: the system can falsely reject an authorised subject, or it can falsely accept an unauthorised subject. Therefore, each system has a False Rejection Rate (FRR) and a False Acceptance Rate (FAR). In the past, the FRR has been disregarded as FAR is much more important for privacy and integrity issues. In elections, availability is (because of the universal requirement) as important as other properties. From a cost and user-friendliness point of view it makes a difference wether systems are already deployed or need to be introduced for the election. Large-scale biometric infrastructures do in general not yet exist.

Combination of Different Techniques

Often, a combination of the above listed authentication techniques is used. The most popular ones are the combination ownership/knowledge and ownership/property respectively in the context of smart and signature cards. The application of these combinations maximises the security because in both cases it is hard to fake the card. Moreover, forwarding the card means giving someone else the opportunity to legally sign any document. In the case of the biometric properties, a vote buyer cannot use a voter's card because he cannot enable the card without the biometric data from the voter.

Result of the Analysis

The advantages and disadvantages are discussed for all three main authentication techniques as well as for mixes. An overview of the result is displayed in Fig. 2.3. As there is no best solution, it needs to be analysed for a particular election which technique should be applied depending on the parameters security, usability, and costs.

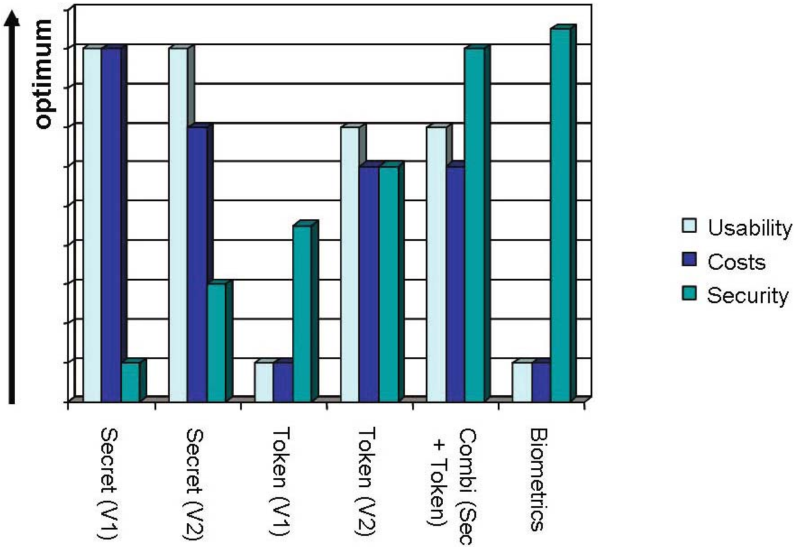


Fig. 2.3. Authentication techniques and their applicability for elections

2.4.2 Techniques to Ensure the Secrecy of the Vote

The challenge to prevent any link between the voter and his (unencrypted) vote can be solved in different ways. In this section, the different approaches to ensure the secrecy of the vote are presented. These approaches can be categorised according to the election phase[9] in which the mechanisms are applied (this categorisation is introduced in [121] and also used in [85]). Table 2.2 shows an overview of the considered categories.

[9] An other way to categorise the approaches is whether the vote or the voter's ID are hidden.

Table 2.2. Categories of anonymisation techniques for electronic voting

Phase	Sub-Categories
Election Setup Phase	Randomised Authentication Token
Polling Phase	Blind Signature (e-vote or authentication token) Separation of Duty Benaloh's Model
Tallying Phase	Mix Net Homomorphic Encryption Hardware Security Model

Anonymisation in the Election Setup Phase

A very simple and non-cryptographic possibility to ensure the secrecy of the vote is to distribute anonymous election tokens to voters. The voter can use this election token to authenticate as an eligible voter without the system knowing who he is.

Anonymisation during the Polling Phase

There exist three different ways to implement anonymisation during the polling phase.

Blind Signature. A blind signature scheme is a method to digitally authenticate a message without knowing the content of the message. Electronic blind signatures work similar as physical blind signatures. Physical blind signatures can be made with an envelope, white paper and carbon paper: something secret is written on the white paper, next the carbon paper is placed on top of the white paper in the envelope, and the envelope is sealed. Next, the so called validator signs the envelope. Obviously the signer does not know what he has signed but on the secret document is his valid signature. The cryptographic variant of blind signatures has been introduced by Chaum in [28]. Here, a message m is first blinded by multiplication with a random number b called the blinding factor (instead of putting the message into an envelope). The result is digitally signed by the validator while the validator has no way of knowing what the original message looked like (instead of the handwritten signature on the envelope). The owner of the original message m can remove the blinding factor b from the signed message $Sig(b * m)$ by dividing out the blinding factor. He gets a valid digital signature for the original message from the validator:

$$Sig(b * m)/b = Sig(m),$$

where m is the original message and b the blinding factor

The implementation for a voting protocol is possible in two different ways: either the e-vote itself is blinded (the most popular variety) or an authorisation token is (like in [82]):

- Blinded e-votes: the voter sends a blinded e-vote to the voting server (often called validator) together with some identification and authentication information. This validator inspect the voter's right to vote. If the voter has the right to vote, the voting server digitally signs the blinded e-vote and sends it back to the voter. The voter now un-blinds the received data and gets a signed e-vote. Then he sends this signed e-vote to a second voting server (often called tallier). The tallier uses the signed e-vote to verify that the e-vote was signed by the validator. Then the tallier knows that the e-vote was sent by an eligible voter, but he cannot decide which voter sent the vote.
- Blinded authentication tokens: the voter sends a blinded anonymous authentication token (instead of the blinded e-vote) to the validator together with some identification and authentication data. He receives a digital signature from the validator on this blinded token. In the next step, the voter computes the value for the signed authentication token and sends this data together with his e-vote to the tallier, which accepts the vote because of the digitally signed authentication token.

The most popular voting protocols implementing blind signatures to ensure the secrecy of the vote are [29, 48, 111, 112], and [80]. An example for a remote electronic voting system implementing blind signatures is the one provided by e-voting.at from the Vienna University of Economics and Business Administration (for more information about this system see [82] and [120]), SENSUS [41] and EVOX [65] (links to the Web pages of the corresponding systems are given in Sect. B.1).

Separation of Duty. The separation of duty approach also works with at least two voting servers, one inspecting the right to vote and another one storing the eligible e-votes. The voter authenticates himself to the first server. In case that he has the right to vote, he receives a random number generated by this first server. This random number is also sent to the second voting server but without any information about the voter's ID. Now the voter uses this random number to authenticate himself as an eligible voter to the second voting server to which he sends his vote in the next step. Again this second voting server can only check whether an eligible voter sent the e-vote but not who.

An example of a remote electronic voting system implementing the secrecy of the vote on a separation of duty basis is the POLYAS system (for more information about the POLYAS system see B.1).

Benaloh's Model. Benaloh's model proposed in [11] is based on a homomorphic secret sharing scheme: each voter shares his vote among n voting servers. The shares are encrypted with the public key of the receiving voting server. At the end of the election day each voting server adds all the received shares to get a share of the election result. Finally the shares of election results are combined to get the total election result.

Anonymisation during the Tallying Phase

This paragraph discusses voting systems and voting protocols which keep the link between a voter and his encrypted vote until the end of the polling phase and publish this link. For instance, the encrypted vote is digitally signed by the voter and published on a bulletin board. As soon as the tallying starts the anonymisation process is triggered. This can be implemented in the following three different ways:

Mix Net. Mix nets have been introduced by Chaum in [27] as a cryptographic alternative to an anonymous channel. It secures who is communicating with whom and it secures the content of the transferred messages. In a Mix net every network participant (or at least a sufficiently large subset of participants) sends encrypted messages to a Mix node (server with special properties) that processes these messages in the following way: a Mix node receives a batch of encrypted messages, decrypts each message with its own secret key, randomises the order in the batch and then outputs the batch of permuted messages to the designated recipient. This is done in a way that the input and output messages are unlinkable (non-deterministic encryption schemes are applied).

Using only one Mix node, it is necessary to trust this particular component not to keep the information about the link between input and output messages. To reduce trust whole Mix networks with n Mix nodes are implemented and incoming messages are encrypted with the public key of each Mix node (in reverse order). Messages are decrypted, shuffled, and forwarded from one Mix to the next one. Now, all Mix components need to collaborate in order to reveal the links between input and output messages of the MIX network - consequently, only one out of n Mix nodes needs to be trustworthy.

As in any other system implementing the anonymisation during the tallying phase, the anonymisation mechanism is applied in the tallying phase. To do so, the encrypted e-votes (without any voter information) are sent through a Mix network. Assuming the Mix network works correctly, the decrypted output votes cannot be linked to the encrypted input votes (and, thus, the voters).

Voting protocols based on Mix networks are presented in [1, 2, 27, 30, 49, 54, 60, 70–72, 77, 103, 116, 129], and [31].

Homomorphic Encryption. Homomorphic encryption is a special form of encryption where one can perform a specific algebraic operation on the plain text values by performing a (possibly different) algebraic operation on the cypher text values:

$$Enc(a + b) = Enc(a) * Enc(b), \text{ for values } a \text{ and } b, \text{ where } + \text{ and } * \text{ denote operations.}$$

In a special type of homomorphic cryptographic primitive it holds that the sum of encrypted values is equal to the encrypted sum of the values. The RSA and ElGamal encryption systems themselves are homomorphic. But in normal use, additional randomness is introduced to enhance the security of RSA and ElGamal. However, this is a desirable property if one wishes to implement an remote electronic voting protocol. Here, it can be used to compute the election result without revealing the content of each encrypted vote because

$$\sum vote_i = Dec(\sum Enc(vote_i)) \text{ holds.}$$

The secrecy of the vote is ensured because single votes are never decrypted by the system. Thus, it does not matter that the link between voter and encrypted vote exists. However, it is important to protect the corresponding decryption key. It is often proposed to distribute the key to n parties, thus, again n or k out of n need to cooperate in order to decrypt single votes (detailed information about secret sharing and threshold cryptography in general can be found in [136] and [32]). The main disadvantage of corresponding voting protocols is that they reduce flexibility, as the votes are essentially limited to yes/no values. Papers have been proposed to overcome this restriction (see, for instance, [39]).

One of the first voting protocol based on homomorphic encryptions is proposed in [34]. An improved and modified version of this voting protocol is presented in [11]. Other voting protocols based on homomorphic encryption can be found in [8, 12, 38, 39, 66, 67, 91, 96, 128, 133], and [92]. Electronic voting systems ensuring the secrecy of the vote based on homomorphic encryptions are VoteHere and the CyberVote system (for more information on these systems see B.1).

Hardware Security Modules. A Hardware Security Module (often abbreviated to HSM) is a tamper-resistant or at least tamper-evident hardware component that can securely generate and store long term secrets for use in cryptography. Generally, it is used to generate a digital key pair without revealing the private key. The revealed public key is sent to the voter who uses it to encrypt his vote. Most HSM systems can also perform further cryptographic operations.

A HSM implementing decryption can be seen as a function which takes as input the encrypted e-votes and returns as output the decrypted result, while the decrypted votes are not revealed.

The Estonian remote electronic voting system uses such an HSM but only to decrypt votes, that is, the tallying software sends encrypted e-votes to the HSM and receives the corresponding decrypted e-votes (see [106]). The sum is computed outside the HSM. Here, the decryption key is protected but the key to activate the HSM needs to be shared-analogous to the decryption key for homomorphic schemes in order to ensure that malicious key holders do not decrypt vote by vote with the HSM (and thereby compromise the secrecy of the vote).

Result of the Analysis

The different approaches to ensure the secrecy of the vote are presented together with a list of protocols and systems implementing the corresponding approach. The advantages and disadvantages are discussed. As there is no best solution, it needs to be analysed for a particular election which technique should be applied.

2.4.3 Client-Side Voting Software

The client-side voting software is essential for the voter to communicate with the voting server. The client-side voting software runs on the vote-casting device. It can either be a specific application or a Web browser, while depending on the system a particular voting applet can be run by Web browsers. Correspondingly, it can be distinguished between a *fat-client*, a *thin-client* (the applet solution), and a *Web browser solution*. These three are discussed in this section with respect to security, usability, and maintenance issues (for a more detailed analysis see [6]).

Web Browser Solution

One possibility to enable the voter to communicate with the voting server is the application of available Web browsers (without a specific client-side voting software). This approach does not involve any kind of (java) applet. Due to the poor security functionalities of a Web browser, the main security mechanisms run on the voting server. The Web browser is only used to establish the link to the voting server, to display the voting Web page, and for the voter to interact with the voting server (authentication of the voter and vote casting). The only assumed security functionality is the Secure Socket Layer (SSL). This is necessary to secure the communication because Web browsers do not provide another possibility to encrypt or sign messages. Using SSL, it is possible to ensure confidentiality and integrity of the exchanged messages. Moreover, the authenticity of the voting server can be ensured (with the help of the voter who needs to check the voting server's certificate).

From a usability point of view, the Web browser solution is welcomed because the voter does not need to install additional software but can use the environment he is used to. Moreover, Web browsers are executable on other devices than normal PCs or notebooks; the voter can also use his WAP mobile phone or PDA to cast his vote using the remote electronic voting system. Concentrating the whole functionality on the server-side has two more advantages: first of all, in the case of a voting system update there is no effort for the voter because only the server-side voting software needs to be updated. Secondly, if a new Web browser or a new version of an existing Web browser is deployed the server-side voting software can be patched in order to support this new Web browser as well. Thus, the voter is free to choose the Web browser he prefers[10] to cast his vote.

[10] Even more, the voter needs to be free to choose a Web browser he wants to use; a particular one cannot be mandated by the responsible election authority (this is caused by the principle of a free election).

However, there are two main disadvantages: first, the remote electronic voting system has no possibility to check the trustworthiness of the vote-casting device, for example, whether there is a virus or Trojan Horse on the vote-casting device which affects the communication between the voter and the voting server. Moreover, an (un-patched) Web browser could weaken the trustworthiness by well-known exploits. The second disadvantage is caused by the poor Web browser functionality. Thus, most of the proposed voting protocols cannot be implemented because they require security functionality on the client-side. For the same reason, this approach can only be used in combination with secrets as authentication techniques. The only disadvantage from the usability point of view is the necessity for the voter to check the certificate of the voting server. This might be new for many voters even if they use SSL on a daily base.

Fat-Client Solution

This approach is called fat-client because the client-side voting software is rich of security functionality and cryptographic algorithms. Such a client-side voting software needs to be installed and executed on the vote-casting device in order to cast a vote.

Any available voting protocol can be implemented using the fat-client approach, thus, in contrast to the Web browser solution, this solution does neither exclude any voting protocol nor any authentication technique. In addition, a fat-client can include a virus scanner or similar security software[11] in order to verify the trustworthiness of the vote-casting device before starting the vote casting process (as proposed in [74]).

The disadvantages of this approach are the distribution, installation, and maintenance of the client-side voting software. Distribution and maintenance is an economic question while the installation is assigned to usability issues. Moreover, the client-side voting software might only run on a particular system if corresponding system properties are given (for example, the java virtual machine is running). Analogously to the validation of the server certificate in the Web browser approach, the voter needs to verify the integrity and authenticity of the voting software he installed on his vote-casting device (this might be for voters more complicated than the verification of a certificate).

Thin-Client Solution

The Web browser solution is from the usability and maintenance aspect preferable while from a flexibility and security point of view the fat-client is advantageous. A mix of both strong points is provided by the thin-client approach. It implements a java applet running in the Web browser. This java applet is the client-side voting software which provides the necessary security functionality on the client-side.

[11] Note, the voter needs to agree and may also not want voting software searching his file system for viruses or the like (for fears about privacy or the federal trojan horse).

Result of the Analysis

The advantages and disadvantages for all three main client-side voting software approaches are discussed. As there is no best solution, it needs to be analysed for a particular election which technique should be applied depending on the parameters security, usability, and maintenance issues. An overview of the analysis is displayed in Fig. 2.4.

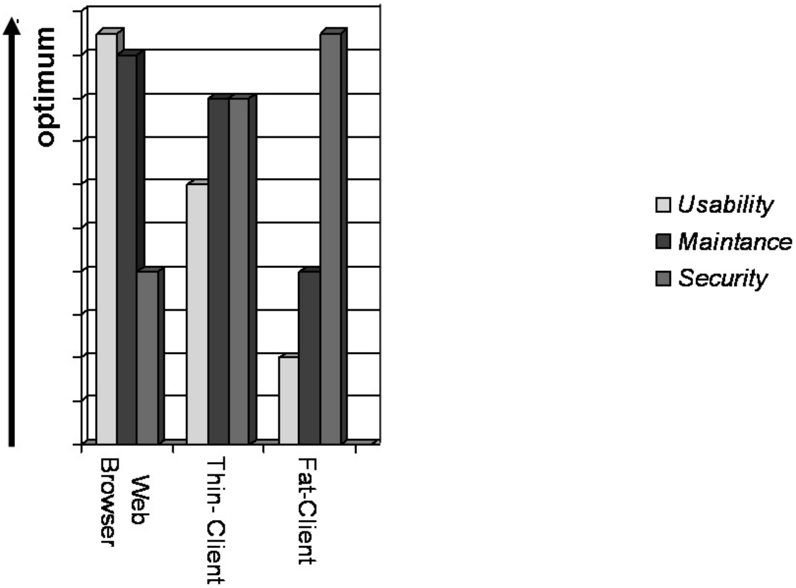


Fig. 2.4. Approaches for the client-side and their applicability for elections

2.5 Summary

This chapter provides fundamentals of electronic voting. An essential contribution is the classification among the three dimensions (medium, environment, and point in time) in Sect. 2.1. In particular, this section shows that the electronic version of the traditional polling station election is constituted by the application of electronic voting machines in the polling station (either stand-alone, networked, or paper-based) and that the electronic version of postal voting is remote electronic voting. As, mostly, the traditional election form runs in parallel to the new electronic implementation. Afterwards, the challenges for multiple channel elections are discussed. The main challenges concern the prevention of multiple vote casting using different channels and the computation of intermediate results. All these findings are essential for the requirement definition, as the requirements depend on the traditional election type to be replaced by the electronic voting system and to which the electronic system should run in parallel.

In Sect. 2.2, the general differences between paper-based and electronic voting are presented. The differences mainly address the possibilities for manipulations: while in the paper-based system it is easy to manipulate the system in general, the electronic voting system enables large scale manipulations. Besides this, the trust aspect plays an important role in this section: while the voters in a paper-based election only need to trust the poll workers, the e-voters also need to trust the developers and administrators.

It is essential for the requirement definition to identify these differences because they require additional requirements for electronic voting systems (meaning requirements that do not exist for paper-based elections).

Section 2.3 focuses on two forms of electronic voting machines and describes technical possible implementation for these types of electronic voting systems.: (stand-alone) direct recording electronic voting machines and the Digital Election Pen system are discussed. As requirements for the Digital Election Pen are already defined in [158], this book focuses on stand-alone direct recording electronic voting machines.

Section 2.4 concentrates on remote electronic voting systems. Different implementations are discussed for the general architecture, the voter authentication, the protection of the secrecy of the vote, and the client-side voting software. The advantages and disadvantages of all implementations are discussed. There is no best solution but it is shown that the implementation depends on many aspects, including the level of the election.

Thereby, the processed and structured information essentially contributes to the definition of requirements for stand-alone direct recording electronic voting machines and for remote electronic voting systems.

3

Related Work – A Landscape of Requirement Catalogues

While the different electronic voting systems are proposed and discussed in the first part of the foundation, this chapter presents an overview and analysis of existing approaches for the evaluation of electronic voting systems. It is discussion is necessary to know these approaches and their vulnerabilities in order to provide an exhaustive list of requirements and an evaluation approach.

The surveyed approaches include requirement catalogues, ordinance, laws, and research activities. The discussed list of requirements were developed by people from different disciplines, like a group of security experts, data protection officers, security auditing enterprises, lawyers, or security auditing civil services.

The first part of this chapter concentrates on requirement catalogues for electronic voting machines (in particular, the German and American election regulations) while the second part discusses those for remote electronic voting systems (in particular, the Council of Europe recommendations, the catalogue for "Online-Voting Systems for Non-parliamentary Elections", the catalogue of the Gesellschaft für Informatik, the Swiss and Austrian election law, as well as the Network Voting System Standards). Afterwards, scientific papers are analysed, in particular Shamos' commandments, Mercuri's PhD thesis, a technical report from the EU CyberVote project, and McGaley's PhD thesis. In all three cases, the analysis is structured according to

- context / background in which the requirements have been developed,
- input sources used,
- type of electronic voting system addressed ,
- categories in which the requirements are classified / level of detail for the requirements,
- proposed evaluation and certification techniques (including underlying trust model), and
- people identified to oversee the evaluation and certification.

The vulnerabilities are summarised in the conclusion.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 37–57, 2009.

3.1 Regulations for Electronic Voting Machines

3.1.1 German Federal Ordinance for Voting Machines

Background. In 1975, the first version of the Bundeswahlgeräteverordnung (BWahlGV – Federal Ordinance for Voting Machines) [143] was integrated into the Bundeswahlgesetz (BWahlG – German Federal Law on Elections). These regulations did allow until recently the use of electronic voting machines in Germany for Federal and European elections[1]. While, the original regulations only addressed mechanical devices, the newest version (which dates back to 1999) extends the list of permitted electronic voting machines to include electronic and software based systems.

Sources. Probably the regulation bases on the Dutch regulations [139]. However, this is not made explicit in the document.

Type of Electronic Voting System. The regulations address *stand-alone electronic voting machines* which are not connected to the Internet or any other network and which are used in polling stations (see Sect. 2.3.1). The devices are used to cast, store, and count the votes while the voter authentication and the inspection of the person's right to vote is done manually by the poll workers.

Requirements. The regulations distinguish between organisational, certification, and technical requirements. The organisational ones mainly define how to deliver the electronic voting machines on election day, what the user-guide must look like, and how to check whether the electronic voting machine is the one that has been evaluated and approved. The technical requirements are defined in the first appendix. Here, the necessary evaluation materials from the manufacturer are defined (this includes the source code). The requirements are divided into two categories: 'technical assembly' and 'functionality'. The 'technical assembly' part is divided in the following sub-categories: construction, resilience, permanency/functional security, reaction, absence of energy supply, and transportation. The 'functionality' part is divided in functional principle, function check, ballot display/appliances, vote storage/tallying/display, sealing, and locking of the devices, vote casting, and ergonomics/usability. These technical requirements are very detailed but at the same time very specific and in some points over-specified; that is they can only be applied to the electronic voting machines in mind.

Evaluation/Certification. The Federal Ordinance for Electronic Voting Machines defines the responsibilities for (re)evaluation, certification, and revocation, but not the evaluation methodology itself, such as the evaluation techniques in use, the evaluation depth, or the underlaying trust model. Some information about the required evaluation can be read out of the necessary

[1] The Federal Constitutional Court decided on March 3th 2009 that the Federal Ordinance for Voting Machines is unconstitutional (compare to [21]).

evaluation material: for instance, the fact that the source code is required might have the consequence that it should be evaluated. There is one evaluation report available [117] but it also does not give any information about the applied evaluation techniques.

Person in Charge. The evaluation needs to be performed by the Physikalisch-Technische Bundesanstalt (PTB – Department of Metrological Information Technology in the National Metrology Institute) and the approval certified by the Federal Ministry of the Interior. But the Federal Ministry of the Interior is currently renewing the regulations and one discussed change is the integration of the Bundesamt für Sicherheit in der Informationstechnik (BSI – Federal Office for Security in Information Technology).

3.1.2 Election Law of the Free and Hanseatic City of Hamburg (Germany)

Background. The state government of the Free and Hanseatic City of Hamburg, Germany was planned to introduce a new type of electronic voting machines for its state parliament (in German: "Bürgerschaft") election in February 2008: the Digital Election Pen. The idea came up because of a change in their local electoral law which causes the use of ballot booklets instead of one side ballot sheets and, thus, results in a time and capacity intensive task for tallying. The persons in charge of the state parliament election proposed a new way to evaluate and certify the Digital Election Pen system because the "Federal Ordinance for Electronic Voting Machines" is not applicable to the Digital Election Pen system. Advised by the Bundesamt für Sicherheit in der Informationstechnik (BSI -Federal Office for Information Security), the person in charge decided to go with the Common Criteria (for more information about this methodology see Sect. 7.1). To do so they contract the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI -German Research Center of Artificial Intelligence) to develop a corresponding Protection Profile [158]. This Protection Profile has been successfully evaluated by the accredited laboratory TÜV Informationstechnik GmbH and certified by the BSI. Nevertheless, the persons in charge decided to contract the Physikalisch-Technische Bundesanstalt (PTB – Department of Metrological Information Technology in the National Metrology Institute) for a second additional evaluation to benefit from their experiences with the evaluations of electronic voting machines. As a foundation the person in charge developed together with the PTB regulations for electronic voting machines [124] corresponding to the Federal Ordinance for Voting Machines. Recently the persons in charge decided not to use the Digital Election Pen system due to negative press and security reservations.

Sources. Hamburg's Regulations for Electronic Voting Machines [124] is based on the Federal Ordinance for Voting Machines in [143] and the election laws and regulations for traditional elections of the city of Hamburg. The Protection Profile [158] was influenced by the parallel work on a Protection Profile for remote electronic voting[2] [161] as well as by the result of a couple of meetings with the persons in charge and the authors of the Protection Profile.

Type of Electronic Voting System. The Digital Election Pen belongs to the paper-based electronic voting systems and here in particular to the optical scan systems. It is described in Sect. 2.3.2.

Requirements. The Hamburg's regulations for Electronic Voting Machines [124] concentrate on the evaluation of functional requirements while the developed Protection Profile [158] concentrates on the security functions deduced from possible threats and policies and it bases on assumptions about the environment. These assumptions are also part of the PTB evaluation. The level of detail in the requirement definition of regulations is compared to the level in the federal ordinance. The requirements defined in the Protection Profile are based on the Common Criteria security functional requirement components.

Evaluation/Certification. The Protection Profile demands an evaluation of the Digital Election Pen system according to the EAL3 (evaluation assurance level) augmented under the trust model defined by the set of assumptions to the environment. The evaluation process is defined in the Common Evaluation Methodology (CEM) [36]. In addition, the Digital Election Pen system is evaluated against the Hamburg's Regulation for Electronic Voting Machines. Similar to the Federal Ordinance for Voting Machines, Hamburg defines the responsibilities for (re)evaluation, certification, and revocation, but not the evaluation process itself, such as the evaluation techniques in use, the evaluation depth, or the underlying trust model.

Person in Charge. The evaluation and certification is done in a cooperation of four institutions in accordance with Hamburg's Election Law for Local Election [52]: Hamburg's Department of the Interior (instead of the Federal Ministry of the Interior) approves the evaluation performed by PTB. Additionally, the BSI certifies an evaluation of the security requirements performed by a Common Criteria accredited laboratory.

3.1.3 American Election Regulations

In the United States, there is a shared responsibility among the three levels of government in overseeing the conduct of elections. Each state sets its own guidelines for the conduct of local, state, and federal elections. States have generally delegated the authority to conduct elections to smaller subdivisions, such as counties, cities or towns. As a result, there are thousands of jurisdictions that administer federal elections throughout the country. However, states must comply with requirements set forth in certain federal legislation in order to receive funding for electoral matters. The most important standards are:

[2] This Protection Profile is called GI/BSI/DFKI Protection Profile and is discussed in Sect. 8.2.

- The Federal Election Commission (FEC) formulated a suggested standard for electronic voting machines in 1990 - the so called Voting System Standard (VSS) [45], but they lacked enforcement authority. The standard was only accepted by two third of the states.
- The Help America Vote Act (HAVA)[3] mandates federal standards for the functionality, accessibility and security of voting systems across the country, as well as for allocating funds to states to help upgrade outdated equipment. HAVA is not exclusively an electronic voting standard; it addresses other types of voting. HAVA established the U.S. Election Assistance Commission (EAC[4]). The EAC's Technical Guidelines Development Committee (TGDC) developed – in cooperation with the National Institute of Standards and Technology – a voluntary guidelines for voting systems, called Voluntary Voting System Guidelines (VVSG) [142]. The VVS guidelines are currently only a draft while the authors ask on their Web page for comments to improve them. They are separated into three parts: part 1 addresses equipment requirements, part 2 documentation requirements, and part 3 testing requirements. Recent discussion by the committee concentrated on the inclusion of mandatory Voter Verifiable Audit Trails and recounts thereof. The main idea for evaluation and certification is that testing equipment for conformance is performed by qualified companies (referred to as an Independent Testing Authority) that are selected by the National Association of State Elections Directors.
- The Institute of Electrical and Electronics Engineers (IEEE) developed an evaluation standard for election voting systems. The purpose of their project (P1583) is to "provide technical specifications for electronic, mechanical, and human factors that can be used by manufacturers of voting machines or by those purchasing such machines. The tests and criteria developed will assure equipment: accessibility, accuracy, confidentiality, reliability, security and usability" [165]. Their detailed report is non-binding but could eventually be incorporated into election system legislation. One group of security experts outside of P1583 was developing a Protection Profile that was expected to be used in the Security Section of P1583. However, this has not been completed, no information is available and it is not clear whether that group is still working on it.

[3] Although HAVA is legally limited to federal elections, in practice it influences virtually all elections in the US. It addresses requirements for electronic voting such as: testing, certification, decertification, and recertification of voting system hardware and software. Also, voting system standards and requirements are addressed (in Sec 301).

[4] The HAVA set up the EAC, a new commission whose responsibility it was to distribute money for updating voting systems and voting administration as well as updating the FEC 2002 Voting System Standards with the assistance of National Institute of Standards and Technology (NIST).

3.2 Requirements for Remote Electronic Voting

3.2.1 Council of Europe Recommendations

Background. In early 2003, the Council of Europe set up a working group to develop a set of standards for e-enabled voting that would reflect member states' differing circumstances. The standards [37] were published in 2004. The correct title is 'legal, operational and technical standards for e-voting - Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe on 30 September 2004 and explanatory memorandum'. Within 112 requirements, the Council of Europe's recommendations are the most comprehensive collection of requirements. The document starts with five recommendations and a list of definitions for election related items. Concrete requirements are then defined in the following three appendices and explained in detail in the "Explanatory memorandum" chapter. This last chapter contains a paragraph "Risk analysis – methodology" where the authors propose the application of the Common Criteria methodology [35] to describe the assets which need to be protected, threats which attack these threats, and corresponding security requirements to protect the threats (here called security objectives according to the Common Criteria). Therefore, they define a long list of assets to be protected, a list of subjects involved and threats which need to be prevented. Based on these threats corresponding security objectives are defined. Even though the work is not completed in the Common Criteria, this parts enables the development of a Common Criteria Protection Profile for these recommendations.

Sources. The group involved in developing these requirements claims to base their results on obligations and commitments from existing international instruments and documents, such as: the Universal Declaration of Human Rights, the International Covenant on Civil and Political Rights, the United Nations Convention on the Elimination of All Forms of Racial Discrimination, the United Nations Convention on the Elimination of All Forms of Discrimination against Women, the Convention for the Protection of Human Rights and Fundamental Freedoms (ETS No. 5), in particular its Protocol No. 1 (ETS No. 9), the European Charter of Local Self-Government (ETS No. 122), the Convention on Cybercrime (ETS No. 185), the Convention for the Protection of Individuals with Regard to Automatic Processing of Personal Data (ETS No. 108), Committee of Ministers Recommendation No. R (99) 5 on the protection of privacy on the Internet, the document of the Copenhagen Meeting of the Conference on the Human Dimension of the Organization for Security and Co-operation in Europe (OSCE), the Charter of Fundamental Rights of the European Union, and the Code of Good Practice in Electoral Matters. The recommendation covers political elections and referendums.

Type of Electronic Voting System. The recommendations address any electronic voting system involving the use of electronic means in at least the casting of the vote and in particular remote electronic voting (defined as "e-voting where the casting of the vote is done by a device not controlled by an election official" [37]).

Requirements. The recommendation of the Council of Europe provides a very comprehensive list of requirements. The document distinguished between legal standards (covering the four election principles of universal, equal, free, and secret suffrages), procedural safeguards (classified in transparency, verifiability/accountability, and reliability/security), operational standards (including the categories: notification, voters, candidates, voting, results, and audits), and technical requirements (containing accessibility, interoperability, systems operation, security, audit, and certification; while security requirements are further classified with respect to election phases and audit requirements with respect to appearing actions like recording and monitoring). In addition, the Election Markup Language (EML) [44] (a standardised XML language for the interchange of data among election services) is recommended.

In 2006, McGaley and Gibson [100] produced a critical analysis of those standards, including a redrafting of the standards themselves in an attempt to overcome some of the drawbacks they had identified in the original. Their analysis "has shown, the CoE standards document is flawed. The inconsistency, incompleteness, over- and under-specification, redundancy and repetition that have been demonstrated could lead to 'bad' systems being certified against these requirements, and/or 'good' systems failing."

In 2007, Rössler used the recommendation of the Council of Europe (in particular the list of security requirements/objectives of the last chapter) as basis to evaluate his proposed remote electronic voting system called EVITA in his PhD thesis "Electronic Voting over the Internet – an E-Government Speciality" [126]. However, he argued that this list needs to be extended. To do so, Rössler applied selected elements of the GI/BSI/DFKI Protection Profile for remote electronic voting[5] [161].

Evaluation/Certification. Requirements 111 in the recommendations states the need for certification process definitions but without giving any details about how such certification could be done. The only information given is in requirement 24 and 25: requirement 24 states that the "components of the e-voting system shall be disclosed" for evaluation purposes. Requirement 25 states that the evaluator has to show that the system "is working correctly and that all the necessary security measures have been taken". However, in order to decide about necessary measures the underlying trust model needs to be defined because this is not yet addressed in the recommendation.

Person in Charge. The recommendation itself does not define the persons in charge of the evaluation or certification. It only states in requirement 25 that the evaluation should be done by an independent body, appointed by the responsible election authority.

[5] This Protection Profile is discussed in Sect. 8.2.

3.2.2 Online-Voting System Requirements for Non-parliamentary Elections

Background. A catalogue of requirements for “Online-Voting Systems for Non-parliamentary Elections” [62] has been developed by the Physikalisch-Technische Bundesanstalt (PTB - Department of Metrological Information Technology in the National Metrology Institute) within the project “Development of concepts for testing and certification of online voting systems” funded by the former German Ministry of Economics and Labour. It has been discussed in two working groups: “Testing and certification” and “Legal framework conditions” ([62] provides a list of groups and people involved in these working groups). The catalogue constitutes a recommendation for developers of electronic voting systems and gives an orientation for the refinement of test concepts. This catalogue is only a recommendation without mandatory regulation character.

Sources. The following already available sources have been considered: the Federal Ordinance for Voting Machines [143], the reports of the CyberVote project [47], the Voting System Standards [45], the Network Voting System Standards [104], and the Swiss Regulations [166] (here especially part 6a: “Pilotversuche mit elektronischer Stimmabgabe” addressing electronic voting).

Type of Electronic Voting System. The electronic voting systems the PTB considered are networked electronic voting machines in polling stations. Remote electronic voting is explicitly not included in the definition. In addition, the authors focused on non-parliamentary elections such as for staff and council work elections as well as shareholder elections.

Requirements. It contains technical and organisational requirements. The requirements are of a sufficiently general level to be described independently of particular systems. The level of detail used in the definition of the requirements is different. The requirements are classified according to the different time intervals or classified as “cross-sectional functions”: preparation of election (including preparation of register of voters, provision of means for voter identification and authentication, preparation of ballot, installation of voting system up to and including readiness for service), polling phase (including voter identification and authentication, management of the register of voters, ballot handling, vote transmission, and vote storage), determination of election result (including termination of vote casting and vote tallying), wrap-up, and safe-keeping (including dismantling and disassembly of voting system, (long-term) archiving, safe-keeping, and maintenance of voting system), and cross-sectional functions (including general reliability of software and hardware, communication system underlying the voting system, anonymisation of votes, and technical observation of voting system (technical audit)).

Evaluation/Certification. The catalogue does not describe “any method to be used for meeting the requirements. It is not even prescribed whether particular requirements are to be met by technical measures or by non-technical operational measures". However, requirement CF3-4 requires that the "implementation shall be proved to be correct with respect to the theoretical concept by software test methods (including code inspections) which represent the state of the art." This is a first hint but no concrete evaluation instruction.

With respect to the trust model , requirement CF2-6 states that the system should comply "with the state of the art in relation to the threat potential accepted". Similar in requirement CF3-2, it is demanded that "the concept used including the mathematical methods shall be appropriate for the particular election." These requirements go in the right direction. However, it is not clear how to define the threat potential and how it influence the evaluation and the required security functions of the electronic voting system.

Person in Charge. Missing evaluation procedures lead to the fact that no persons in charge are identified to run the evaluation or even certify any systems.

3.2.3 Catalogue of the Gesellschaft für Informatik

Background. The Gesellschaft für Informatik (GI - the German society of computer scientists) presently has about 24.000 members mainly from Germany. There are also associated memberships in Austria and Switzerland. It was set up in Bonn in 1969. The rules for elections of the bodies of the GI are formally specified by the GI's regulations for elections and polls [50]. Since July 2003, article 3.5.4 of the constitution of the GI allows the application of remote electronic voting. Here, the precondition is that the remote electronic voting system provides the same security level as postal voting. In all cases where postal voting is admitted the election committee can decide to also give members the possibility to use a remote electronic voting system - as long as it is comparably secure. In the summer of 2004, the chairmanship (in German: "Präsidium") decided unanimously to offer both postal voting and remote electronic voting for the chairmanship elections in December 2004. The GI established a group of security experts to accompany the pilot election and the future process of remote electronic voting in the GI. The group consists of German experts in IT security and electronic voting from universities, the Physikalisch-Technische Bundesanstalt (PTB - Department of Metrological Information Technology in the National Metrology Institute), and the executive board of the GI. The main task of the expert group was to develop and enforce ad-hoc security requirements. In December 2004, the Internet voting expert group of the GI decided to develop a requirements catalogue for "Internet-based elections in societies" [113]. The catalogue should be short and crisp and should not exceed six printed pages. After several iterations, the last version was published in 2005.

Sources. Four requirements catalogues were already available and were used as a basis for further development: the Council of Europe recommendation [37], the IEEE Voting Equipment Standards [165], and the PTB requirements "for Online-Voting Systems for Non-parliamentary Elections" [62].

Type of Electronic Voting System. The GI requirements address remote electronic voting using secrets (voting TAN) for authentication.

Requirements. The catalogue starts off with some preliminary notes and explicates assumptions under which any applied Internet voting system must ensure the security requirements. For example, it is assumed that a non-secret name or a membership number (user-id) is applied for the voter identification and a secret alphanumeric password (one-time election PIN) is used for voter authentication. Moreover, it demands in these preliminary notes that the electronic ballot box and the electronic election register are installed on different servers and that the two servers are located in different organisations. This part is very specific compared to other requirement catalogues. The preliminary notes also define issues which are out-of-scope of the security requirements catalogue. For example, the candidate nomination and the maintenance of the list of eligible voters are not considered in the catalogue. Rules for long-time storage of the election results are not addressed, either. The catalogue of 2005 separates the requirements on the system development and on the election execution from the requirements on the remote electronic voting system itself. The requirements on the remote electronic voting system itself are divided into requirements on the election servers and on the election software. The general requirements for system development contain requirements on the type and level of details of the system description, the security analysis and the manuals. There are especially strong requirements on the anonymity concepts. This category includes requirements on the development process, the system tests, and the key management. The requirements on the election execution contain the distribution of the election PIN, the election register management, and the installation as well as the de-installation of the voting system. The catalogue requires the election servers to run a secure operating system, and to isolate the election software from all other applications. Only authorised persons may have access to the servers. For the requirements on the election software the following categories were used: general requirements to a remote electronic voting system and its security, specific functional requirements to the remote electronic voting system, requirements with respect to the anonymity of votes, specific requirements to ensure a universal and equal election, and ergonomic and usability requirements. The general functional requirements include the system's reliability and logging as well as the guarantee of consistent system states in the case of any interruption. Specific functional requirements refer to the electronic register and to the electronic ballot box. Requirements with respect to anonymity specify a secret, equal, and universal election. The last category of requirements on the election software addresses ergonomic and usability issues.

Evaluation/Certification/Person in Charge. The document does not talk about evaluation and certification procedures nor does it talk about the underlying trust model or the person in charge.

3.2.4 Swiss Election Law

Background. The Swiss political system can be described as a direct democracy, meaning each voter has at least four times per year the possibility to cast a vote for referenda on the national, cantonal, and communal level. Moreover, in all cantons[6] postal voting is allowed without any conditions and it is copiously used. Introducing remote electronic voting is seen as a possibility to simplify the processes and decrease the costs. The project "vote electronique" is a consequence of the Swiss strategy to use the new information and communication technologies for the decision making process. Thus, Switzerland started running three pilot projects: in Genève, Neuchâtel, and Zurich. Their notion of remote electronic voting includes casting a vote in elections, referenda, electronic signature of initiatives, requests for referenda and candidate proposals. In order to enable legal binding trials on a federal level, the federal law regulating political rights [53] was changed together with the corresponding regulations [166].

Sources. From the law it is not deducible whether or which previous available resources defining electronic voting requirements were used for the forming of this law.

Type of Electronic Voting System. The Swiss projects address remote electronic voting where the user can use any kind of device connected to the Internet.

Requirements. Art. 27a-27q of the ordinance of May 24 in 1998 on political rights [166] contains the requirements which must be ensured before the Federal Council can approve pilot trials of remote electronic voting. The Swiss requirements can be summarised as follows: "e-voting has to be as secure and reliable as the traditional voting methods (that is, postal voting and voting at polling stations" [15].

The main regulations are addressed in article 6 a of [166] (Art. 27 a -q). Art 27 (a) – (d) and (q) regulate how pilot projects have to be permitted and be set up in general. The other parts are not clearly structured for people having a technical background. However, the headings of the following parts are: (e) Protection of the formation of options (for example, by enabling the voter to change his choice before he finally casts it), (f) encryption (for example, encryption of the vote before it leaves the voter's device), (g) secrecy of the vote (for example, by demanding that it is not possible to link vote in the electronic ballot box), (h) further mechanisms to ensure the secrecy of the

[6] There is one exception – the canton Tessin, which does not implement postal voting unconditionally, but only for elections and referenda on the cantonal level.

vote (for example, that the voter needs to be informed on how to delete all vote related data from his PC), (i) control of the right to vote, (j) one-voter-one-vote principle, (k) securing cast votes (l) technical state of the art, (m) computation of the election result, (n) solve technical problems, and (o) check the efficiency (meaning analysing the turnout and voter behaviour). There is one big difference between the Swiss regulations and others: the Swiss regulations do not demand that any attack must be prevented but only systematic ones, which seems to be more realistic. As the requirements are formulated in the regulations, they are rather abstract and less technical. Thus, for the developer it is hard to decide which security functionality is sufficient to meet these requirements.

Evaluation/Certification/Person in Charge. Art. 27(l) of the Swiss Election Law demands that the enforcement of the security requirements and the functionality of the electronic voting system needs to be approved by an independent external authority, which is accredit by the Swiss federal chambers (in German: "Bundeskanzlei"). The same holds for changes within the electronic voting system. There is no statement about the evaluation methodology to evaluate a system against the defined requirements. This is left up to the evaluators.

3.2.5 Austrian Election Regulations

Background. In Austria, electronic voting is allowed for student union elections (see [19]) and for the Austrian Federal Economic Chamber (in German: "Wirtschaftskammer") elections (see [18]), since 2001. Both regulations are very similar, in particular, they both require the Austrian citizen card (in German: "Bürgerkarte") for voter identification and authentication.

Sources. From the law it is not deducible whether or even which previous available resources defining electronic voting requirements were used for the forming of this law.

Type of Electronic Voting System. It is not clearly defined what kind of electronic voting system is addressed. Nevertheless from the regulations it can be deduced that Austria wants to either apply remote electronic voting or a kiosk system while in both situations the identification has to be done by the citizen card.

Requirements. The regulations are very short but are embedded in a broad environment of information technology applications. §34 (4) [19]/§74(2) [18] demands that the electronic voting system needs to be compliant with the security objectives for digital signatures according the signature law [20] and the data protection law [17]. Moreover, it also contains the general demand that electronic voting must be as secure as the traditional system (§48 (2) [19]). In §34(5) [19]/§74 (3) [18] some more technical requirements are defined: 1./(a.) ensuring the secrecy of the vote, including that no one can link the voter to his vote at any point in time; 2./(b.) checking the voter's right to vote before he will see the ballot; ensuring the one-voter-one-vote principle; 3./(c.) integrity of cast ballots by the application of digital signatures; secrecy of the vote during transmission by encryption; 4./(corresponding regulation in §74(2) [18]) all possible actions of the responsible election authority are also possible with the electronic voting system; 5./(d.) preventing accidentally cast votes; 6./(e.) providing a polling booth (in case of electronic voting machines). Moreover, according to §48 (2) [19] electronic voting is only allowed to be applied in parallel to a paper-based system. In §78 (6) [18] there is an additional organisation requirement defining that the electronic voting process needs to be stopped if it does not work correctly anymore. For a more detailed discussion see [126] (section 2.3 and 5.2).

Evaluation/Certification. In § 34 (6) [19]/§74 (4) [18] the evaluation procedure is addressed: the state of the art of the used system should be sufficiently and permanently scrutinised. But, it is not stated how this should be done.

Person in Charge. – Furthermore, §27 (6) [19]/§74 (4) [18] demands that the compliance with the security requirements needs to be certified by a certification authority according to §19 of the signature law.

3.2.6 Network Voting System Standards

Background. The Network Voting System Standards (NVSS) are proposed in [104] and have been developed by employees from the VoteHere company (they also retain the copyright in [104]) which developed its own electronic voting system. In parallel to writing these standards the Federal Election Commission of the US (FEC) revised the Voting System Standards [45]. One of their tasks was to include standards for public network direct recording electronic voting systems but explicitly no other online or network voting systems outside the polling station. VoteHere sees their Network Voting System Standards as both, an alternative and as input to the FEC work and to ensure that upcoming trials of remote electronic voting and kiosk electronic voting machines "are conducted using systems that have been evaluated and demonstrated to meet a set of standards sufficient to protect the integrity of the election" [104].

Sources. The Network Voting System Standards are based on two technical reports from VoteHere ([163] and [164]). Moreover, the standards are based in part on the Voting System Standards [45] and on the findings and recommendations of the SERVE report [73], the CalTech/MIT Voting Technology Project, a workshop on remote electronic voting and on private research efforts at VoteHere.

Type of Electronic Voting System. The NVVS are intended to be applicable to any electronic voting system which transmits votes over a network and which is not under the physical and logical control of the election officials at all times. This includes remote electronic voting and kiosk electronic voting machines.

Requirements. VoteHere distinguishes in their standards between high-level and functional requirements as well as specific standards. From the high-level requirements the functional ones are deduced. The functional requirements shall be met by all systems regardless specific architectural division between hardware, firmware, and software underlying technology or implementation methodology. They are organised around the four high-level requirements: fairness, accuracy, privacy, and proof[7]. The definition of the demanded standards is distinguished in hardware, software, telecommunication, cryptographic, quality assurance, and configuration management standards. While the functional requirements are discussed in detail these standards cover only one page.

All requirements are qualified by the words "shall" or "must" and in addition they are identified through the use of an ID, a unique alphanumeric number. They also provide background information on some of the requirements for better understanding. Requirements relating to the electoral register are outside the scope of these standards.

Evaluation/Certification. – The Network Voting System Standards propose to start with a design review. In case the design is logically able to meet the requirements, the election result of the first design evaluation provides the necessary details for specific functional review and testing. The evaluation begins in accordance with the NVSS with an examination and review of the technical data package. This includes a check of whether all necessary documentations for the further steps are available and the review of the quality assurance and configuration standards. In the next step the design is reviewed and afterwards there are two steps to be done in parallel: code review and hardware tests. The last step contains system functional testing. Certification processes are not addressed.

Person in Charge. – The standards do not talk about persons in charge for evaluation or certification.

3.3 Scientific Papers

Almost all scientific papers proposing a voting protocol are structured in the following way: the authors start with a set of requirements, then they describe their proposed voting protocol and then show in the analysis part that their system ensures the previously defined requirements. First of all, these requirements are only related to the voting protocol and secondly, it is not that surprising that the protocol ensures its own defined requirements. Thus, such papers are not taken into account for this discussion. This section concentrates on work independent of concrete voting protocols or electronic voting systems. A selection of the most important contributions is discussed in this section:

[7] "Proof – The system must, without violating the privacy requirements, be able to prove that the fairness and accuracy requirements have been met" [104].

Shamos' commandments [137], the PhD thesis from Rebecca Mercuri [101], the list of requirements provided in the CyberVote project [47], and the PhD thesis from Margaret McGaley [99].

(A) **Shamos Commandments**

Background. The work around [137] is based on the author's participation in official evaluations of about fifty different electronic voting systems since 1980 as well as an audit of the election laws of about half of the United States.

Sources. From the paper it is not deducible whether or even which previously available resources defining electronic voting requirements were used for the forming of the commandments.

Type of Electronic Voting System. Shamos does not further limit the implementation of electronic voting. He addresses any electronic voting system that captures and tallies votes.

Requirements. In [137], system requirements for electronic voting are boiled down to the following six high level commandments:

1. "Thou shalt keep each voter's choices an inviolable secret."
2. "Thou shalt allow each eligible voter to vote only once, and only for those offices for which she is authorised to cast a vote [...]".
3. "Thou shalt not permit tampering with thy voting system, nor the exchange of gold for votes."
4. "Thou shalt report all votes accurately."
5. "Thy voting system shall remain operable throughout each election."
6. "Thou shalt keep an audit trail to detect sins against Commandments II-IV, but thy audit trail shall not violate Commandment I."

While 1)-3) are strong ones, 4)-6) are more flexible ones from the author's point of view and the first one is the most important one. Auditing is not part of the commandments because the author argues that "no existing voting system is auditable" [137].

Evaluation/Certification. Evaluation is addressed in [137] by suggesting testing to show that tampering is not possible, but that it is discouraged and difficult. The statement in this paper is that electronic voting systems that meet these six commandments should be certified for use in public elections. However, a particular methodology for the testing is not proposed neither is the impact of different trust models taken into account. In addition, he does not talk about a formal certification process.

Person in Charge. The author does not talk about person in charge to run the evaluation and certification procedures.

(B) Electronic Vote Tabulation – Checks and Balances (Mercuri's PhD Thesis)

Background. Mercuri proposes, in her PhD Thesis "Electronic Vote Tabulation – Checks and Balances" [101] besides other important issues, the application of the Common Criteria methodology to evaluate existing and proposed electronic voting systems. From her point of view, "the establishment of generalised PPs for voting system requirements, therefore, is viewed as an essential base for the development of consistent policies under which evaluation of proposed voting systems can be performed"[8] [101]. With this statement she is first to recommend the application of the Common Criteria for electronic voting and in particular for electronic voting machines. However, she only provides basic discussions about the applicability of the Common Criteria but did not start developing a Protection Profile.

Type of Electronic Voting System. She discusses in her thesis various types of electronic voting machines in polling stations, while concentrating on lever machines and direct recording electronic voting systems.

Requirements. The prosed requirements contain the following categories: system requirements, functionality, correctness (accuracy), accountability, disclosability, reliability, integrity, availability, fault tolerance, data requirements, confidentiality, retention, and recountability, user requirements, administrator requirements, interface usability, documentation, testing, paths, facility management, recovery, system distribution, and compliance with laws and regulations.

Evaluation/Certification. By proposing the Common Criteria methodology, the evaluation and certification procedure is appointed to the Common Evaluation Methodology (see Sect. 7.1 for more information on the Common Criteria). In this context she discusses the evaluation depth. Mercuri proposes the Common Criteria evaluation assurance level EAL4 as the lowest level that should be applied to certify electronic voting systems, "since all lower levels omit salient requirements involving the development process. EAL4 does not include any covert channels analysis, which first appear in EAL5, so perhaps the higher level should be used as the minimal assurance evaluation standard [...] Since the attack potential of the voting system is likely to be high, the more stringent EAL6 evidence of resistance should also be included" [101]. However, as her thesis only serves as first step, she did not discuss possible trust models as part or the Common Criteria evaluation.

Person in Charge. She does not explicitly name persons in charge, but one may suppose that she – according to the CC – suggests the evaluation to be done by an accredited testing authority and the certification to be done by corresponding CC authorities.

[8] PP means Protection Profile in the Common Criteria. For more information see Sect. 7.1.

(C) Voting System Requirements in the CyberVote Project

Background. The list of requirements in [47] has been developed in the EU CyberVote project which is a research and development (RDT) programme funded by the European Commission under the fifth framework programme (FP 5). The objective of the project was to develop a highly secure voting prototype which can be used for remote electronic voting (using a PC or mobile phone). The project is carried out by a consortium led by MATRA Systemes & Information and a grouping together of British Telecommunications, NOKIA Research Centre, K.U.Leuven Research & Development, Technische Universiteit Eindhoven, Freie Hansestadt Bremen, Mairie d'Issy-les-Moulineaux, and Kista Stadsdelsnämnd.

Sources. The list of requirements results from discussions among the CyberVote consortium and from various interviews with responsible election authorities and electronic voting experts from Germany, France, and Sweden.

Type of Electronic Voting System. – The project addressed remote electronic voting while different kinds of authentication techniques have been used in different trials and the developed voting protocols [133] use homomorphic encryption in order to ensure the secrecy of the vote.

Requirements. In [47], the authors distinguish between user requirements and functional specification. The first class contains those system requirements from a user's perspective (VOT) (users are the voters, responsible election authority, and the service providers), meaning those functions required to support the user tasks and the user-interfaces. The second set is classified in two categories: legal requirements (LEG) and technical requirements (TEC). It is distinguished in general requirements and those addressing specific national issues or sometimes reflect different ways of approaching remote electronic voting. They are all uniquely identified.

The authors make an interesting point with respect to the list of user requirements: their development is an ongoing process because in the beginning users may not appreciate the benefits that an innovative system can offer them; but once they understand the benefit of a new technical solution, their requirements may change.

Evaluation/Certification. The document recommends the evaluation of the system by different parties: national experts and software experts. Moreover, they propose interviews and usability tests with potential voters and mathematical proof for the correctness of the system. However, a detailed evaluation instruction as well as the incorporation of the trust model is missing.

Person in Charge. The authors do not talk about person in charge to run the evaluation and certification procedures.

(D) E-voting: An Immature Technology in a Critical Context (McGaley's PhD Thesis)

Background. McGaley explores in her PhD Thesis "E-voting: an Immature Technology in a Critical Context" [99], besides other important issues, two approaches to develop requirements for e-voting (top-down starting with the Recommendations of the Council of Europe [37] and bottom-up like in this book) and discusses the evaluation of systems. For the Recommendations of the Council of Europe, she reveals flaws and problems and then improves the requirement document. The following considerations are only done for the bottom-up approach.

Sources. It is based on the requirements listed in the German Regulations for Electronic Voting Machines [143], the requirements defined in the recommendations of the Council of Europe [37], and the "Online-Voting Systems for Non-parliamentary Election" catalogue developed by the Physikalisch-Technische Bundesanstalt (PTB – Department of Metrological Information Technology in the National Metrology Institute) [62].

Type of Electronic Voting System. "As these requirements were developed for critical elections, remote e-voting systems are not considered [...]. Similarly, it is assumed that the election devices are not networked (data can only be transferred between election devices by physically moving some storage medium). We exclude voter-registration and voter-authentication from our current analysis [...]; we assume that they are implemented as per paper-only elections. [...] The requirements in their current form are not flexible enough to cover non- DRE e-voting systems. [...] Therefore, this catalogue excludes, for example, mark-sense (also known as optical-scan) and digital election pen systems" [99].

Requirements. The catalogue is divided into security, functional, usability, organisational, assurance, audit system, and VVAT requirements (where VVAT means Voter Verified Audit Trail).

Evaluation/Certification. In [99] the following evaluation methodologies are proposed: usability testing, including "sociology-style experimentation with suitably representative test subjects", election observation to evaluate whether organisational, assurance and VVAT requirements are met, manufacturer compliance tests, code review, functionality testing, end-to-end testing, and environmental testing by an independent testing authority as well as red team testing (meaning penetration tests). It is not discussed how the testing should happen, how deep the evaluation should be and on which trust mode the evaluation should base.

Person in Charge. Person in charge of the evaluation is mainly named by independent testing authorities. The responsible election authority is in charge of the system certification.

3.4 Result of the Analysis

Within the proposal of existing requirement catalogues, their vulnerabilities are pointed out for each catalogue. These identified vulnerabilities can be categories in three classes, namely those related to the requirement definition, those addressing the underlying trust model and those concerning the evaluation and certification process. According to this classification, the vulnerabilities can be summarised as follows:

- The specified lists of *requirements* – the electronic voting system needs to ensure – differ in the level of detail and with respect to their focus. The different levels of detail occurs also inside one document. In particular the laws define rather high level requirements, while other documents formulate a set of more detailed and more technical requirements. Some of the requirements are over specified, others under specified or too abstract. Sometimes, documents contain contradicting requirements. There are also cases where the defined requirements are tied that much to a particular electronic voting system that it is impossible to apply such requirements to any other electronic voting system. Moreover, for a better understanding a clear definition of important terms is missing.
- With respect to the definition of a *trust model*, a (detailed) description of the limiting factors is missing or at least not made explicit in the analysed documents.
- The applied *evaluation methodology* is not considered in most of the documents. Thus, evaulators could concentrated on the evaluation of the voting protocol or the development process, the electronic voting system as a whole, the user-interfaces, or the robustness against unexpected events. What kind of test are required is also not defined (for instance, penetration, functional, black box tests).

  Furthermore, a concrete statement about the evaluation depth is missing: is a source code analysis of the whole system or of important parts of the system required or is the evaluation of the high level design of the electronic voting system enough.

Consequently, it is not obvious how to decide for most of these catalogues why a system should pass or fail an evaluation. Moreover, some group of experts could decided that a particular system is secure enough while other would not recommend to use this system. Note, the only exceptions are those dealing with the Common Criteria as evaluation methodology. However, this approach has been proposed and discussed several times but has never been completely implemented[9].

The above critic does also hold for projects where no such catalogues were available or used; and the responsible election authoritys, caused by their

[9] The only exception is the Protection Profile for the Digital Pen. However, this is based on the same approach as proposed in this book and is written by the same author (as one of two authors).

non-technical background, were supported in their decision making process concerning whether a particular electronic voting system is "secure (enough)" (for instance in the Estonia and Dutch remote electronic voting project and in the Irish electronic voting machine project). Here, experts were asked to evaluate particular electronic voting systems (but without specifying the evaluation methodology). Thus, depending on the experts' background and knowledge as well as the project set-ups, including time and money constraints, the evaluation process differs in all three relevant aspects: underlying requirements, the trust model, and the applied evaluation methodology.

Additional Outcome. The analysis of existing literature also shows that the list of requirements depends on the type of electronic voting system in mind. From a high level point of view the requirements are the same, while as soon as more technical requirements needs to defined, it is necessary to know whether the electronic voting system in mind is a remote electronic voting system or the Digital Election Pen.

3.5 Summary

This chapter elaborates on existing requirement documents for electronic voting to learn from their vulnerabilities and to provide an exhaustive list of requirements in this book.

In the category of documents for electronic voting machines the German Federal Ordinance for Voting Machines, the Hamburg Regulations for Voting Machines, the Protection Profile for the Digital Election Pen, and the American approaches are presented and analysed according to the proposed structure (in Sect. 3.1). The requirement documents for remote electronic voting systems discussed in Sect. 3.2 are: the Council of Europe recommendations, the PTB catalogue, the GI list of requirements, the Swiss and Austrian regulations, as well as the Network Voting System Standards. The discussed scientific work from Sect. 3.3 contains Shamos' commandments, the Mercuri's PhD thesis, the CyberVote requirements, and McGaley's PhD thesis. Many of these documents refer to each other or even form the bases for each other. Figure 3.1. illustrates this relationship.

Section 3.4 discusses the vulnerabilities of the analysed documents which mainly address the different levels of detail for the requirement definition, the missing introduction of a trust model, and the absence of concrete guidance for the evaluation, including a statement about the evaluation depth. Additionally, the analysis shows that the list of requirements depends on the type of electronic voting system in mind.

The requirement documents presented in this chapter are used as input for this book to develop an exhaustive list of requirements for electronic voting which overcomes the identified vulnerabilities. In order to demonstrate the exhaustiveness character, the new list of requirements proposed in Chap. 5

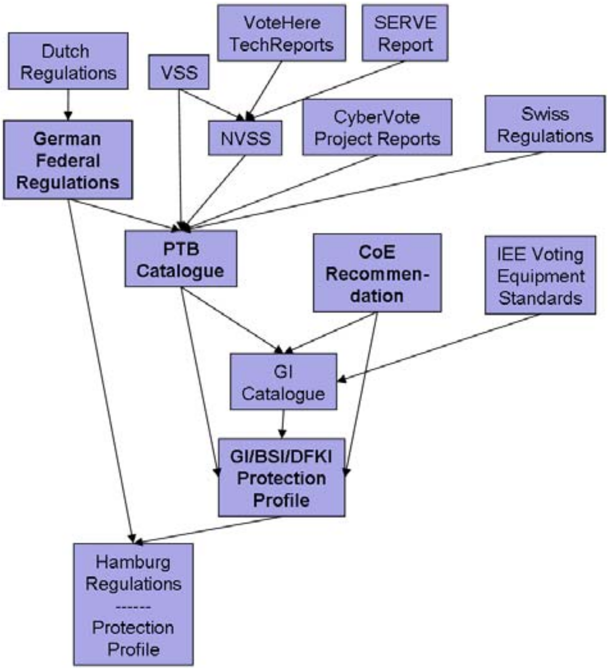


Fig. 3.1. Relation between different requirement catalogues

and 6 refers to the corresponding requirements in existing documents. Due to time and place constraints, the provided requirements only refer to a subset of these documents. These are the following ones:

- The Council of Europe Recommendations [37] because it contains the most comprehensive and exhaustive list of requirements
- The Federal Ordinance for Voting Machines [143] because it has already been applied for system evaluations
- The PTB catalogue for "Online-Voting Systems for Non-parliamentary Elections" [62] because it results from a research project with a huge advisory board

Part II

Requirements

4

Process and Framework Description

The first part of this book builds the foundation for the requirement definition by presenting, classifying and discussing different implementations of electronic voting and by discussing and analysing available requirement documents for electronic voting systems. Here, the advantages, disadvantages, problems, and vulnerabilities of each document are identified. Based on these fundamentals, a new list of requirements is developed in the second part of the book. This list combines and improves the existing requirement documents from Chap. 3. Strictly speaking there is not one list but two, one for stand-alone direct recording voting machines and one for remote electronic voting systems.

The fourth chapter describes the process to obtain the final list of requirements. The development process includes the development of a glossary, the definition of syntax and semantics, a detailed description of the target of evaluation, the development of a first draft of requirements, the incorporation of existing literature, the provision for the election principles, the defence of identified threats, and the classification in different sub-lists. In order to work with election principles and threats, both concepts are explained in the second part. It is important to know this process to understand why it is claimed that the list is standardised, consistent, and exhaustive.

There are three aspects of electronic voting which are not further discussed in this book: election observation, verifiability, and vote updating. These aspects are neither taken into account for the definition of requirements nor for the evaluation methodology. As these aspects are important and might be included in an extended version of the later framework, their main ideas and challenges are discussed in this chapter.

4.1 Description of the Procedure

The list of requirements as presented later in this book is developed within the following six phases:

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 61–72, 2009.

Phase 1: Definitions and Notation

Within existing requirement literature it is sometimes difficult to determine the exact intended meaning of a requirement because the words used can be interpreted in different ways. In order to provide an unambiguous meaning to terms used in the requirement definition, an election glossary has been developed (see appendix C for the glossary). Additionally, the syntax and semantics applied for the requirement specification have been defined (see Sect. 4.4 for the definition of syntax and semantics). The glossary as well as the syntax and semantics are essential to provide a *standardised and consistent list of requirements* that is easy to understand and where each requirement has an unambiguous meaning.

In this early phase, the decision was made to leave out organisational requirements that are already used in traditional elections (for example, for electronic voting in polling stations, poll workers must confirm a voter's identity and his right to vote before authorising him to vote; for more examples see appendix D).

Phase 2: Target of Evaluation

Before starting to develop the requirements, the targets of evaluation have been defined. The analysis of the existing literature in chapter 3 shows that the definition of one single list of requirements that works for all forms of electronic voting system as defined in Sect. 2.1 is not feasible. While some requirements are common for all types, many are different from type to type. In addition, the discussion in Sect. 2.1 shows that the requirements (at least in detail) depends on the traditional election type which should be replaced by the electronic voting system or to which the electronic system should run in parallel. Thus, a clear definition of the target of evaluation and its environement is essential. In the following chapters, this book focuses on the following two forms of electronic voting for the requirement definition:

- Stand-alone direct recording electronic voting machines[1] (see Sect. 5.2 for the exact definition of the target of evaluation) and
- Remote electronic voting systems (see Sect. 6.2 for the exact definition of the target of evaluation).

Based on this decision, the requirements for stand-alone electronic voting machines have been designated to one list, and those addressing remote electronic voting systems have been designated to a second list.

[1] The list of requirements for stand-alone electronic voting machines results from a cooperation with Margaret McGaley from the Department of Computer Science at the National University of Ireland in Maynooth. A first draft, which is enhanced for this book, has been published in [156].

Phase 3: First Draft of Requirements

An initial list of requirements for stand-alone direct recording electronic voting machines has been developed based on the author's own experience and understanding of electronic voting systems as well as flaws and vulnerabilities found in existing requirement catalogues (see chapter 3). This draft made use of the glossary and terminology developed in phase 1. The goals for this draft were

- to provide one sentence per requirement,
- to keep sentences simple,
- to explicitly address the subject by starting the sentence with it,
- to maintain the same level of abstraction for all of the requirements, and
- not to repeat the mistakes from the existing literature, that is, being too concrete (over-specified) and, thus, excluding potentially good electronic voting systems or being too abstract (under-specified) and, thus, making it impossible to decide if a particular system meets the requirement or not.

The following findings have been identified by developing this first draft:

- Certain terms can only be defined within a given context (for instance, "election data"). Such terms cannot be further defined but are identified in the corresponding requirements as so-called *responsible election authority variables*.
- Some of the requirements are interconnected. Making these connections explicit with cross-references helps to develop a *consistent list of requirements*. These references are not highlighted in this book. However, this information can be found in [99].

The following two steps (phase 4 and 5) have been taken to improve this first draft of requirements for stand-alone direct recording electronic voting machines. The result is published in [156]. This list has been extended and adapted for remote electronic voting systems by running through phase 3 to 5 again. In parallel, the list of requirements for stand-alone direct recording electronic voting machines as published in [156] has also been enhanced during this process.

Phase 4: Improvement Based on Existing Literature

After having developed a first draft, the following three catalogues of requirements from Chap. 3 have been chosen for comparison to the draft list for its enhancment[2]:

- Requirements listed in the German Regulations for Electronic Voting Machines [143],
- Requirements defined in the recommendations of the Council of Europe [37], and

[2] In future work, the other discussed requirement catalogues from chapter 3 should also be crosschecked.

- Requirements proposed in the “Online-Voting Systems for Non-parliamentary Election” catalogue developed by the Physikalisch-Technische Bundesanstalt (PTB - Department of Metrological Information Technology in the National Metrology Institute) [62].

The goal of this comparison is to provide an *exhaustive list of requirements* by ensuring that all requirements mentioned in these three catalogues have either already been listed in the draft list, are added, or left out for good reasons. Requirements that are left out and correspondent reasons are discussed in appendix D (for instance because the requirement is over-specified). To validate this completeness-statement the requirements from [143], [37], and [62] are linked to the requirement in the list presented in this book.

Note, the requirements for stand-alone direct recording electronic voting machines in Chap. 5 refer to the corresponding requirements listed in these three catalogues. The requirements for remote electronic voting in Chap. 6 only implicitly refer to them by referring to the corresponding requirement for stand-alone direct recording electronic voting machines. However, those requirements that are mentioned in these three catalogues that only fit for remote electronic voting are linked directly to the corresponding requirements for remote electronic voting in Chap. 6.

Phase 5: Double Check

In order to further emphasise the completeness idea, two more rounds of improvements have been undertaken to develop the list of requirements as provided in Chap. 5 and 6. First, in terms of the election principles, the legal component has been analysed, and then possible threats are discussed:

- For the first step, the election principles have been analysed in order to verify whether all of their aspects are covered (see Sect. 4.2 for the definition of each election principle). This has been done in particular according to the arguments in [168] and [78] (two legal sources) as well as according to the KORA method described in [61]. If one or more aspects were not covered by the current list of requirements, corresponding requirements have been added.
- In the second step, a list of possible threats has been identified. This has been done based on a threat analysis and in particular a threat tree. The completed list has been checked to determine whether all of the listed threats are prevented by the current list of requirements. If a threat has not been prevented, the list of requirements has been correspondingly extended (see Sect. 4.3 for more information about the definition of threats).

The requirements in Chap. 5 and 6 are labelled with the corresponding election principles and security requirements refer to the threats that they prevent.

Phase 6: Classification

As a last step, the requirements have been classified into the following three categories in order to make the lists more readable:

- Security requirements: system requirements[3] that mitigate against threats.
- Functional requirements: system requirements deduced from organisational policies that specify the behaviour of the electronic voting system.
- Assurance requirements: requirements related to activities during the development and evaluation of the product to assure compliance with requirements.
- Additional requirements: this category contains the following two subclasses:
  – Usability requirements: requirements that are related to user-interfaces.
  – Operational requirements: requirements related to the responsible election authority and poll workers.

4.2 Election Principles

In order to ensure within phase 5 that all aspects of the election principles are covered, they are introduced and explained in this section.

Elections that utilise any kind of electronic voting system need to comply with the existing election laws and regulations. Therefore, it is important to take these into account when defining requirements for electronic voting systems. In 2002, the European Commission for Democracy through Law (Venice Commission) adopted a non-binding Code of Good Practice in Electoral Matters[4] in which the following five principles (that is, the so-called election principles) are identified as fundamental: universal, equal, free, secret, and direct suffrage. In [149], it is shown how to deduce technical requirements from these juridical requirements.

Although no generally accepted definitions of these principles exist, their meanings for this book[5] are the following (identical to those presented in [156]):

[3] System requirements are requirements related to the software and hardware of the electronic voting system.

[4] Code of good practice in electoral matters: (Venice Commission – Opinion 190-2002-el), endorsed by Parliamentary Assembly Resolution 1320 (2003) and CLRAE Resolution 148 (2003), subject to a Declaration by the Committee of Ministers (114th session, 13 May 2004).

[5] To compare, the definitions in [37] are the following ones: "universal suffrage: all human beings have the right to vote and to stand for election subject to certain conditions, for example, age and nationality;" "equal suffrage: each voter has the same number of votes;" "free suffrage: the voter has the right to form and to express his or her opinion in a free manner, without any coercion or undue influence;" "secret suffrage: the voter has the right to vote secretly as an individual, and the state has the duty to protect that right;" "direct suffrage: the ballots cast by the voters directly determine the person(s) elected."

Secret: [se] The voting system shall prevent anyone without the appropriate authority[6] from deducing or proving the link between a particular ·*elector*· and his ·*vote*·.

Free: [fr] The voting system shall protect the ·*voter's*· right to express his ·*vote*· in a free manner, without any coercion or undue influence.

Equal: [eq] The voting system shall ensure that each ·*voter*· may only ·*cast*· one ·*vote*· per ·*poll*· [7].

Universal: [un] The voting system shall protect the right of an ·*eligible voter*· to ·*cast*· his ·*vote*·.

Direct: [di] The voting system shall determine the results of a ·*poll*· based on all ·*votes*· ·*cast*· and only based on these ·*votes*·.

In the process of assigning these principles to the requirements it became clear that many requirements could be said to be upholding both 'freedom of vote' and 'direct election', or both 'equal suffrage' and 'direct election'. Thus, the best way to clarify this is to apply the election principles of *freedom of the vote, equal suffrage*, and *universal franchise* to individual ·*voters*·, and *direct election* to collections of ·*votes*· (see Fig.4.1).

elec. setup phase	**polling phase**	**tallying phase**	**archiving phase**
	◁- - - - -	secret	- - - - - ▷
	◁ - - free - - ▷		
		◁- - - - - direct	- - - - - ▷
◁- - - - - equal	- - - - - ▷		
◁- - - - - universal	- - - - - ▷		

Fig. 4.1. Election phases

The following two further categories of principles have been added in order to be able to link each requirement to at least one category: "trust" and "data protection". Trust has more often been assumed as an implicit principle, and the objective is to implement an electronic voting system with the aim of maximising public trust. Data protection is necessary when referring to remote electronic voting because information about the voters are used to identify him in the remote electronic voting system.

[6] In most constituencies, no such authority exists; the U.K. is one notable exception.

[7] In certain ·*polls*·, some ·*voters*· may have the right to ·*cast*· more ·*votes*· than others (for example, stock corporations). Such ·*polls*· are not taken into account for this book.

4.3 Threats

In phase five, threats are identified. Here, several aspects of the intrusion are considered in the threat description: intruder's goal, motivation, and technical capability.

Intruder's goal and motivation. An attacker is assumed to have one of the following goals and motivations for his intrusion:

- Compromising the secrecy of the vote
- Selling the vote / buying votes / force people to vote in a particular way
- Affecting the election result
- Computing intermediate results
- Confusing voters
- Collecting personal data (in the case of remote electronic voting)

Intruder's technical capability. Within the intrusion description the following different capabilities and knowledge are distinguished:

- Outside intruders – those who use any available public information. A subset of outside intruders are ineligible voter or those intruders who can read, add, delete, and alter messages on the network.
- Inside intruders – those who are part of the process, like administrators, poll workers, and software developers but also everyone who has access to the electronic voting system after the election when it is switched off and, thus, cannot protect itself anymore. These people have easier access to the electronic voting system than outside intruders. For example, they could change software/hardware and data in the electronic voting system and can decrypt data because they may know the secret keys.
- Malicious voter – A voter who either tries to cast more than one vote (malicious electors), tries to sell, or to prove how he voted.

4.4 Syntax and Semantics

The list of requirements is written in the following syntax with corresponding semantics[8]:

[8] The requirement definition is based on a new and propriety method and not on the Common Criteria notation (see Sect. 7.1 for the application of the Common Criteria). In this phase, it was not yet decided whether to use the Common Criteria for the evaluation part. However, the applied method is particular suitable because (other than the Common Criteria,) it allows to reference to existing catalogues in a transparent way and to apply the glossary. Moreover, the applied methods partly adopts the Common Criteria. Thus a later translation is easily possible.

- Security requirements, corresponding threats, and functional requirements are labelled in the following way: threats are labelled with ***T.**NameOfThreat*, security requirements with ***O.T.**NameOfRequ*, and functional requirements with ***O.OSP.**NameOfRequ*. The "O" means (security) objective, "T" means threat, and "OSP" means organisational security policy[9]. "T" and "OSP" has been added to highlight that some requirements can be deduced from threats and others from organisational security policies. However, while the threats are explicitly named together with the security requirement that prevents it, the organisational security policies are not made explicit because an organisational security policy belongs to exactly one corresponding functional requirement. Mentioning the organisational security policies would just mean reformulating the functional requirement in such a way that it can be read as a policy.
- The remaining requirements are labelled with numbers and use the following prefixes: **Usab.** for Usability requirements, **Op.** for operational and organisational requirements, and **Assur.** for assurance requirements.
- According to the Common Criteria, the application notes (***Appl. Note:***) have been introduced. These are added to further explain the requirement with which they appear.
- In this document, the key words `shall` and `should` are to be interpreted as described in RFC 2119 [105]; that is `shall` means that the definition is an absolute requirement of the specification whereas `should` means that valid reasons may exist in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

  Reasons vary for choosing `should` instead of `shall` :
  - To ensure compatibility with the law in any known jurisdictions. For example, the U.K.'s election law demands that an appropriate authority must be able to link a voter to his vote (under particular circumstances), which is forbidden in many other countries.
  - To provide a different level of strength for one requirement. For instance, the system `should` be tamper-resistant and `shall` be tamper-evident. In this case, the system has to be at least tamper-evident while the responsible election authority can decide that the system has also to be tamper-resistant, but the system must be at least tamper-evident.
  - To provide "none-core" requirements, that is, those requirements that do not hold for all kind of elections. Those requirements that are identified as non-core are marked with a [non-core] label.
- The requirements refer to the existing catalogues in the following way: CoE x refers to the corresponding requirement in the standards of the Council

[9] According to the Common Criteria, organisational security policy (OSP) means "a set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment." [35]

of Europe document [37]; PTB y refers to the corresponding requirement in the catalogue developed by the Phsikalisch-Technische Bundesanstalt (PTB – Department of Metrological Information Technology in the National Metrology Institute) [62], and BWGV a and BWGV-A1 b respectively which refer to the main part and the appendix respectively of the German Regulations for Electronic Voting Machines [143]. In the case where a requirement in one of the original documents was deemed to cover more than one concept, it was split into sub-standards, which are denoted by letters, with divisions made along natural lines.

- For the assurance, organisational, and usability requirements as well as for the requirements for the audit system, the PhD thesis of Margaret McGaley [99] is additionally taken into account. She also improved the requirements based on the common list provided in [156].
    - Requirements provided in this book which differ from those in [99] are labelled with an asterisk ‘*’. Differences exists in labelled election principle, application of `shall` and **should**, added items in enumerations, differences with respect to responsible election authority variables, or reformulations.
    - Requirements labelled with “**” do not appear in [99][10].
- Words and phrases that are defined in appendix C are formatted ·*like this*· throughout the list.
- The responsible election authority variables appear underlined throughout the requirements.
- In order to show which of the listed requirements is determined by which of the election principles ([se] , [fr] , [eq] , [un] , [di] , [tr] , [dp]), each requirement below is associated with at least one of them. In the case where all principles are supported by a particular requirement, it is marked with [all] .
- Threats are itemised together with the corresponding security requirement. They are labelled with the same label that is used for the corresponding requirement. In the case in which more than one threat is listed for one requirement, these threats have an additional label: “A” and “B” . In general, one or more threats correspond to one security requirement, but no more than one security requirement is assigned to any one threat. There is one exception for remote electronic voting systems – O.T.IntegVotesAterPP and O.T.AuthCheck are linked to T.IntegVotesAfterPP.

  For each threat the type of intruder is made explicit. The goal or motivation of the threat is added after the term “in order to”.

4.5 Beyond the Scope

Election Observation. Observation of elections is done by one or more independent parties (typically from another country or a non-governmental or-

[10] There is no requirement added by [99].

ganisation – NGO), primarily to ensure the fairness of the election process. Public organisations, like the Organisation for Security and Co-operation in Europe (OSCE), the European Union (EU) and the United Nations (UN) are used to observe paper-based systems. They are accustomed to observe the entire process in all election phases.

Nowadays, these organisations are learning how to observe electronic voting. Some research support is given in [88, 162], and [89]. The OSCE, for example, observed stand-alone electronic voting machines in polling stations in the US, the Netherlands, and Belgium (corresponding reports can be found in [46, 107, 108], and [109]). Moreover, in March 2007 they took the next challenge by observing remote electronic voting in Estonia (see [106] for the report). Here, they did not concentrate on the technical detailed issues because of the short observation period, but more on the procedures surrounding the election; for instance, by whom and how was the system evaluated, how was the system set-up and installed, how was it ensured that the proper software had been installed, who has access to the system, is it user-friendly enough, redundancy and availability concerns, are there procedures in the case of failures, etc.

Verifiability. Caused by the lack of transparency and driven by more and more problems with current systems, verification mechanisms are proposed in research literature (see, for example, [112] and [138]). According to [118], verifiability means that there are mathematical algorithms to ensure that the election has been properly conducted. Three different forms can be distinguished:

- *Universal verifiability* meaning that anyone (voter, responsible election authority, or external auditors) can verify the election result after the announcement of the tally. One popular implementation for electronic voting machines is the Voter Verified Audit Trail.
- *Individual verifiability with open objection* to the tally, which is a weaker requirement allowing every voter to verify that his vote has been properly taken into account. In case of a miscounted vote, a sound complaint can be produced, without revealing the content of the vote.
- *Individual verifiability*, which is an even weaker requirement, since it allows for individual voter verification but forces voters to reveal their vote in order to file a complaint.

These mechanisms might be at first glance a welcome concept, but with a closer look most aspects can be identified as counterproductive in practice, as shown in [153]. Moreover, more effort for the voter arises, as part of his voting process is focused on the verification procedure. As verifiability is not yet implemented in practise, this topic is not further discussed in this book. However, requirements for verifiability in electronic voting systems are defined in [167] and [99].

Vote Updating. As discussed in Sect. 2.2, remote electronic voting should only be used in parallel to postal voting because here the disadvantages of casting votes in unprotected remote environments has already been accepted. Nevertheless, remote electronic voting has so many advantages for our highly mobile world that people have investigated solving the problems of family voting and voter coercion for remote electronic voting in unprotected environments in order to implement remote electronic voting in parallel to traditional polling station elections.

The major proposed solution against such fraud is called 'vote updating' (also called provisional voting) in which a voter may cancel any previous cast votes by casting a new vote. Vote updating has already been used in some countries like in Sweden for traditional paper-based elections where voters having cast an absentee vote, but, can cancel this vote by casting a vote on election day in the polling station. Vote updating in the context of electronic voting has become popular by the Estonian elections in 2005 and 2007.

With vote updating, an intruder can still observe the voter or force him to cast a particular vote but the voter has the possibility to cast later on another vote and, thus, to make another choice and cancel the first vote(s). So, it becomes unattractive for an attacker to visit people in order to force them to cast a vote. Moreover, any voter who would like to change an unwanted vote could do so at any time. For the same reason, ballot buying becomes unattractive. In addition, vote updating overcomes the problem that remote voters are in general early voters and cannot respond to short-term political events. In the electronic implementation of vote updating, it is requires that, during the polling phase, e-votes are assigned to the voter who cast them and the time when the voter cast them. At the time of the vote tallying, all votes of a given voter must be gathered and only the one with the most recent time stamp is counted while all others are deleted.

Disadvantages mainly refer to social aspects: with vote updating, there is a risk to lose the graveness and the value of elections. It could become similar to a game or some unimportant polls in the Internet or on TV. Vote updating only increases the security if the voters take the opportunity to cast several votes. Indeed, most of the voters will not do so. In Estonia, they counted 364 of 9681 repeated e-ballots and 30 of them cancelled e-votes by casting a paper vote on election day. Therefore, currently, it might be a nice, technically easy but only theoretical solution, which does not overcome the problems in practice. An exhaustive analysis of vote updating possibilities, advantages, disadvantages, and technical implementations can be found in [145].

4.6 Summary

This chapter describes the development process for the requirements which are specified in the following two chapters. Together with the underlying syntax and corresponding semantics, the standardised, consistent, and exhaustive characteristics of the new lists are ensured.

Therefore, in Sect. 4.1, the six phases of the development procedure for the requirement lists are described: (1) notations, (2) target of evaluation definition, (3) development of the first draft, (4) improvements based on existing literature, (5) improvements based on a discussion of the election principles and possible threats, and (6) a classification of the requirements (in system, assurance and organisational requirements, while the system requirements are further classified in security, functional, and usability requirements). This section also states which two types of electronic voting systems are chosen for the definition of requirements, namely stand-alone direct recording electronic voting machines (used in polling stations) and remote electronic voting systems.

In order to better understand the fifth phase, both aspects involved, namely the election principles and the threats, are discussed in detail: in Sect. 4.2, the election principles are defined and assigned to election phases. Besides the five election principles, two additional categories are introduced: requirements assigned to the category trust and requirements assigned to the data protection law. Afterwards, Sect. 4.3 classifies different intruder's motivations to attack an electronic voting system and introduces and defines different intruder types, including inside and outside intruders as well as malicious voters and malicious electors.

Section 4.4 provides the syntax and corresponding semantics used for the requirement definition. It contains explanations about notations (for instance, for items defined in the glossary), prefixes for the different requirement categories, definitions for particular words such as shall and should, labels for election principles, references to existing literature, specific wordings, and marks.

In Sect. 4.5, election observation, verifiability, and vote updating are discussed as new electronic voting-specific issues. These aspects of electronic voting are beyond the scope of this book as corresponding implementations first need to be further discussed and improved and then integrate them in the requirements.

5

Requirements for Electronic Voting Machines

The previous chapter describes the requirement development process – including quality assurance measurements – and presents the applied syntax and semantics. The results of this development process are two large, standardised, consistent, and exhaustive lists of requirements. This chapter defines the requirements for stand-alone direct recording electronic voting machines. Before providing the list of requirements the chapter-specific notation is explained; meaning those notations used for this chapter but not in the requirements definition for remote electronic voting systems. In addition, the exact target of evaluation under consideration for the requirement specification is defined. Then, the two main subgroups of system requirements - security and functional requirements - are presented separately. Both parts distinguish between requirements for the polling phase and those requirements for the tallying phase. In addition, the list of functional requirements contains detailed requirements for the audit system. The last part specifies the assurance, usability[1], and operational requirements.

5.1 Citation and Additional Notations

The security and functional requirements listed in this chapter represent a further improvement on the requirements that are listed in [156]. In addition to extensions and textual changes, the requirements have been reordered according to section 4.4. In particular this contains the following aspects:

[1] Although usability requirements belong to the category of system requirements, they are discussed in a different section because these requirements are not further treated in the evaluation part.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 73–91, 2009.

- They are separated into security requirements, which are deduced from threats, and functional requirements, which refer to organisational security policies[2].
- The requirements are labelled by names or shortcuts rather than numbers as in [156].

To indicate the relationship between [156] and the requirements listed here, corresponding labels are added by "Paper Sec_x, Funct_y". The security and functional requirements are categorised into the following sub classes: those that need to hold during the polling phase and those that need to be ensured 'only' after the polling phase. The functional requirement subsection also contains a list of requirements for the audit system.

5.2 Target of Evaluation

The requirements below mainly address one particular group of electronic voting systems, namely those called **stand-alone direct recording electronic voting machines in polling stations** (see Sect. 2.1 and 2.3.1), that is, votes are cast and stored on dedicated electronic voting machines that are not networked. The electronic voting machines in mind should be used instead of traditional polling station elections. Corresponding to the definition of a "stand-alone electronic voting machine in polling stations" in Sect. 2.1, voter registration, identification, and authentication is accomplished manually (the same processes and techniques as in traditional paper-based elections in polling-stations). Thus, the considered target of evaluation does not provide voter registration, voter identification, or voter authentication functionality. Corresponding requirements are therefore not considered. In addition, the functionality of the target of evaluation only covers the polling phase and the tallying phase. Thus, the election setup and archiving phase are not addressed in the security and functional requirements. Instead, it is assumed that the electronic voting machines are set up correctly and contain the proper candidate list and the proper definition of valid and invalid votes (that is, in general proper configuration)[3]. In addition, it is assumed that the machines are set up in polling booths. Therefore, requirements to ensure a protected environment are not addressed. The target of evaluation includes the following components:

- The electronic voting machine with the vote-casting interface.
- A connected poll worker interface to enable and disable the vote-casting interface.
- The tallying software. It can either run on the electronic voting machine or on another external device, such as an arbitrary work station.

[2] In [156] the requirements are also categorised into security and functional requirements, but the separation criteria are not clear.

[3] However, one requirement (that is, O.OSP.SelfCheck) demands that poll workers have the ability to check the configuration before starting the polling phase.

5.3 Security Requirements

5.3.1 Security Requirements for the Polling Phase

T.AC: An outside intruder gets access to the ·*electronic voting machine*· without knowing or having the access tokens to tamper the ·*electronic voting machine*· in order to reach any of his goals.

–

O.T.AC [all] The ·*electronic voting machine*· `shall` implement an access control policy which restricts all activities on the ·*poll worker interface*· to particular ·*user*·-roles.

Paper Sec_3

T.Tamper: An inside intruder tampers with the ·*electronic voting machine*·, altering its appearance, behaviour, and/or internal data in order to reach any of his goals (for instance, to affect the ·*election result*· by altering, adding or deleting ·*votes*·).

–

O.T.Tamper [all] The ·*electronic voting machine*· (including the ·*e-ballot box*·) `should` be tamper-resistant. The ·*electronic voting machine*· (including the ·*e-ballot box*·) `shall` be tamper-evident.

BWGV-A1 B(2.1b, 2.4a)
CoE [15, 29, 34a, 80, 86a/c, 92]
PTB VP[1-2, 4-3, 4-4, 5-2b], CF[1-9b]
Paper Sec_15

Appl. Note: The only interfaces to the ·*electronic voting machine*· should be the ·*vote-casting interface*· (including those designed for ·*voters*· with disabilities) and ·*poll worker interfaces*·. Where other interfaces exist they shall be disabled.

T.UnauthVotesA: A malicious ·*elector*· logs on the ·*electronic voting machine*· for a second time to cast another ·*vote*· in order to affect the ·*election result*·.

–

T.UnauthVotesB: An outside intruder adds ·*e-vote*· using other interfaces than the ·*vote-casting interface*· in order to affect the ·*election result*·.

–

O.T.UnauthVotes [di] The ·*electronic voting machines*· `shall` ensure that ·*e-votes*· can only be added through the ·*vote-casting interface*· and only during the ·*polling phase*·.

BWGV-A1 B [3.1b, 3.6a-d]
CoE [5a, 91, 94b, 96a]
PTB VP[1-6, 3-13, 3-17]
Paper Sec_1, 12, 18, 19

Appl. Note: The ·*electronic voting machine*· shall be automatically put in an ·*inactive state*· after the ·*voting process*· is finished. The ·*poll worker interface*· should provide the functionality to put the ·*electronic voting machine*· in an ·*inactive state*·.

T.SepDuty: An inside intruder abuses his access privileges to tamper with the ·*electronic voting machine*· in order to reach any of his goals.

–

O.T.SepDuty [all] The access control mechanism `shall` only allow access to the ·*electronic voting machine*·, if at least two different ·*users*· are logged in.

CoE [33]

T.ElectionSecrecy: An in-/outside intruder gets access to the ·*electronic voting machine*· and uses the stored information to link ·*voters*· to their ·*votes*· in order to compromise the secrecy of the vote.

–

O.T.ElectionSecrecy [se] The ·*electronic voting machine*· `should` not store any information which could link the ·*voter*· with his ·*vote*· after the completion of the ·*voting process*·. Where any information which could link the ·*voter*· to his ·*vote*· is stored on the ·*electronic voting machine*·, it `shall` only be accessible to those with appropriate authority.

CoE [16, 17, 34b, 35]
PTB VP[1-2, 3-15, 5-2a],
CF[1-9c, 3-1, 3-2]
Paper Sec_11

Appl. Note: The ·*electronic voting machine*· shall store the ·*e-votes*· in a history independent way (that is, the ·*vote casting*· order shall not be preserved and no timestamp shall be stored with the ·*e-vote*·).

Appl. Note: According to O.T.Tamper the electronic voting machine shall be tamper-evident meaning tampering can be detected but with respect to the protection of the secrecy of the vote, it is then already too late.

T.AvailInfo: An in-/outside intruder in or close by the polling station sees, hears, or measures information provided by the ·*voting process*· in order to compromise the secrecy of the vote.	–
O.T.AvailInfo [se] During the ·*polling phase*· the ·*electronic voting machine*· `shall` not give any information about the ·*voting process*· outside the ·*vote-casting interface*·, except for the current ·*electronic voting machine*· state (·*active state*· or ·*inactive state*·), the number of ·*votes*· ·*cast*· so far, and feedback according to O.T.NegFeedback. ***Appl. Note:*** The ·*electronic voting machine*· shall prevent any emissions which might endanger the secrecy of the ·*vote*·. This includes any kind of sounds and detectable radio waves. The ·*electronic voting machine*· shall protect the secrecy of the ·*vote*· against power analysis.	Paper Funct_1, 2, 3
T.SecrecyAfterBreakd: An inside intruder with access to the ·*electronic voting machine*· after a ·*electronic voting machine*· breakdown, exception, or malfunction reads the last ·*voter's*· ·*selections*· and/or ·*vote*· in order to compromise the secrecy of the vote.	–
O.T.SecrecyAfterBreakd [se] In case of ·*electronic voting machine*· breakdowns, exceptions, and malfunctions, it `shall` not be possible to link the last ·*voter*· with his ·*selections*· or ·*vote*·.	CoE [16, 19] Paper Sec_9

5.3.2 Security Requirements for the Tallying Phase

T.AffectCounting: An in-/outside intruder installs malware on the machine running the ·*tallying software*· in order to affect the ·*election result*·.	–
O.T.AffectCounting [di] The ·*tallying software's*· operations and data `shall` be unaffected by other applications.	BWGV-A1 B[2.2, 2.5, 3.7c] CoE [26b], PTB CF[1-2, 1-7] Paper Sec_24

T.IntegElecData: An inside intruder tampers with ·*election data*· after the ·*tallying phase*· in order to affect the ·*election result*· in the case of recounts.	–
O.T.IntegElecData [di] The ·*tallying software*· `should` protect the integrity of ·*election data*· (at least including: ·*votes*·, ·*results*·, and audit information) as soon as results are calculated.	BWGV-A1 B[3.4f] CoE [57, 75b, 97] PTB DR[2-7], CF[1-7] Paper Sec_25
T.IntegVotes: An inside intruder tampers with ·*e-votes*· after the ·*polling phase*· and before the ·*tallying phase*· in order to affect the ·*election result*·.	–
O.T.IntegVotes [di] The ·*electronic voting machine*· `shall` protect the integrity and authenticity of ·*e-votes*· as soon as the ·*polling phase*· is closed.	Paper Sec_14
O.T.AuthCheckCount [di] The ·*tallying software*· `shall` verify the integrity and authenticity of ·*e-votes*· before starting the ·*tallying phase*·.	CoE [34c, 86b, 97, 107c] PTB DR[1-3, 2-7] Paper Sec_23
T.LinkInParalElec: An inside intruder with access to ·*e-votes*· after the ·*polling phase*· discovers some aspect of ·*voters'*· identities by examining ·*votes*· that were cast together. For instance, non-citizen residents may have limited voting rights. An intruder could determine which votes came from a particular community.	–
O.T.LinkInParalElec [se] [non-core] The ·*electronic voting system*· `shall` prevent anyone from linking different ·*e-votes*· from the same ·*voter*· to one another (when parallel ·*polls*· are run).	BWGV-A1 B[2.4b] Paper Sec_16

5.4 Functional Requirements

5.4.1 Functional Requirements for the Polling Phase

O.OSP.NeutInter [fr] The ·*electronic voting machine*· and the ·*vote-casting interface*· shall be optically neutral.

BWGV-A1 B[3.3a]
CoE [90a]

O.OSP.EqualPres [fr] The ·*electronic voting machine*· shall ensure equality and accuracy of presentation of ·*voting options*·.

Appl. Note: The ·*electronic voting machine*· shall avoid the display of influencing messages.

BWGV-A1 B[3.3b]
CoE [12, 47, 48]
PTB VP[3-1 – 3-3, 3-5, 3-8]
Paper Funct_4

O.OSP.AccurDisp [fr] The ·*electronic voting machine*· shall accurately display the authentic and unaltered ·*ballot*·.

CoE [90a]
PTB VP[3-1, 3-2, 3-3]
Paper Funct_6

O.OSP.PosFeedback [tr] The ·*electronic voting machine*· shall provide feedback to the ·*voter*· regarding the status of his ·*vote*· (It shall at least contain the information that his ·*e-vote*· has been successfully stored in the ·*e-ballot box*·).

Paper Usab_5

O.OSP.PWClosePoll [all] The ·*poll worker interface*· shall warn the ·*poll workers*· if they try to close the ·*election*· before the final date.

–

O.OSP.Spoil [fr] [non-core] The ·*vote-casting interface*· should provide the functionality for the ·*voter*· to ·*spoil*· his ·*vote*·.

BWGV-A1 A[a]

O.OSP.SpoilWarning [fr] [non-core] The ·*vote-casting interface*· should warn the ·*voter*· when he tries to ·*spoil*· his ·*vote*· in one or more ·*polls*·.

PTB VP[3-9]
Paper Funct_12

O.OSP.StoreAllVotes [di] The ·*electronic voting machine*· shall store all ·*e-votes*· ·*cast*· over the ·*vote-casting interface*· in the ·*e-ballot box*·.

BWGV-A1 A[b,1]

O.OSP.NoInteraction [un] The ·*electronic voting machine*· `shall` prevent ·*voter*· interaction in case of exceptions and malfunctions.

PTB CF[1-12]
Paper Sec_7

O.OSP.Robust [un] The ·*electronic voting machine*· `shall` be robust against power outage, unexpected ·*user*· activities, and environmental effects (for instance, mechanical, electromagnetic, and climatic).

BWGV-A1 B[2.2,2,3,2.5,3.7c]
CoE [30]
PTB CF[1-7, 1-9a]
Paper Funct_13, _14

O.OSP.InfoPW [di] [non-core] The ·*electronic voting machine*· `shall` indicate to the ·*poll worker*·

- the number of ·*votes*· ·*cast*· so far and
- its current state.

BWGV-A1 B[3.4d]
PTB VP[1-7]

O.OSP.V-Interface [fr] The ·*vote-casting interface*· `shall` provide the functionality for the ·*voter*· to

- change his ·*selections*· before ·*casting his vote*·,
- easily cancel his ·*voting process*· at any time, and
- clear all his ·*selections*·.

BWGV-A1
B[3.3c/d,3.6b/d/f]
CoE [11, 13]
PTB VP[3-10b, 3-11, 3-14]
Paper Funct_11

O.OSP.PWInterface [se] [fr] The only functionality provided by the ·*poll worker interface*· is

- starting the ·*polling phase*· (which is only possible once),
- resuming the ·*polling phase*· after breakdowns or other problems (according to O.OSP.ErrorRecovery),
- closing the ·*polling phase*· (after which only the export of data and in particular ·*e-votes*· is possible),
- acting according to messages from O.OSP.NegFeedback, and
- checking the ·*electronic voting machine*· in arbitrary ways (according to O.OSP.SelfCheck).

Appl. Note: The ·*electronic voting machine*· shall not provide any functionality to calculate ·*results*· during the ·*polling phase*·.

BWGV-A1 B[3.4f,3.5c/d/e]
CoE [34a, 53a]
PTB PE[4-10b], DR[1-3, 2-1],
VP[3-17, 4-3a, 5-2a, 5-6]
Paper Sec_13

O.OSP.DeleteRecord [se] Whenever a ·*voter*· completes his ·*voting process*· (by ·*casting*· his ·*vote*· or ·*canceling his voting process*·) any records of his ·*voting process*· `shall` be deleted from display.

CoE [11, 52a, 93a]
PTB VP[3-16]
Paper Sec_10

O.OSP.Availability [un] The ·*electronic voting machine*· `shall` be available during the whole ·*polling phase*·.

Appl. Note: Any backup system shall ensure the same requirements as the main voting system.

CoE [70b]
PTB PE[4-5]

O.OSP.PWCheck [all] The ·*poll worker interface*· `shall` provide the functionality to check that the ·*electronic voting machines*· have been set up correctly (for example, order of ·*voting options*· and empty ·*e-ballot box*·).

PTB PE[4-10a]

O.OSP.LastVote [un] The ·*electronic voting machine*· `shall` provide the functionality to determine whether the ·*e-vote*· of the last ·*voter*· was successfully stored in the ·*e-ballot box*· in case of

- exceptions,
- malfunctions, and
- after ·*electronic voting system*· breakdowns.

Paper Sec_8

O.OSP.ErrorRecovery [di] [un] The ·*voting server*· `shall` run a self-check before s resuming is possible. In the case of irreversible problems the ·*voting server*· `shall` prevent a resuming of the ·*polling phase*·.

PTB CF[2-3]

O.OSP.Auditing [tr] The ·*electronic voting machines*· `shall` be capable of producing comprehensive audit data.

CoE [59, 83a, 102, 104]
PTB PE[4-3b], CF[4-1]
Paper Sec_4

O.OSP.NegFeedback [un] The ·*electronic voting machine*· `shall` provide feedback in the form of error messages in case of exceptions and malfunctions.

BWGV-A1 B[3.2b]
PTB VP[5-5a]
Paper Sec_6

O.OSP.DataLoss [di] The ·*electronic voting machine*· `shall` prevent data loss during normal operations and in case of

- exceptions,
- malfunctions, and
- after ·*electronic voting system*· breakdowns.

BWGV-A1 B[2.3, 3.4e]
CoE [34a, 77, 99]
PTB VP[3-9, 5-3],
CF[1-9a, 1-11]
Paper Sec_2

O.OSP.AccurRep [fr] The ·*electronic voting machine*· `shall` ensure that the ·*voter's*· ·*selections*· are accurately represented in the ·*e-vote*·.

Appl. Note: Some recommend the provision of a voter-verified paper audit trail. But this points is controversial and disputed. This aspect is not further discussed in this book. However, it is addressed in [99].

BWGV-A1 B[3.3c/e]
CoE [95]
PTB VP[4-1, 4-2]
Paper Funct_5

O.OSP.SelfCheck [tr] The ·*electronic voting machine*· `should` regularly perform automatic self-checks while it is in the ·*inactive state*·. The ·*electronic voting machine*· `shall` be capable of performing self-checks.

Appl. Note: The ·*electronic voting machine*· should automatically check that the ·*e-ballot box*· is empty before the ·*polling phase*· begins. The ·*poll worker interface*· `shall` provide the functionality to check that the ·*electronic voting machine*· has been set up correctly.

BWGV-A1 B[3.2a, 3.5a/b]
CoE [72a, 79, 89b]
Paper Sec_20, _21

O.OSP.CompatClient [all] [non-core] The ·*electronic voting machine*· `should` be compatible with other devices (such as those used by people with disabilities) where appropriate.

CoE [64, 66, 67, 68]
Paper Funct_10

O.OSP.AdequNoVotes [un] [non-core] The ·*electronic voting machine*· `shall` be capable of recording an adequate number of ·*votes*·.

BWGV-A1 B[3.4a]
Paper Funct_8

O.OSP.AdequNoBallotOpt [fr] [non-core] The ·*electronic voting machine*· `shall` support an adequate number of ·*voting options*·

BWGV-A1 B[3.3e]
Paper Funct_9

5.4.2 Functional Requirements for the Tallying Phase

O.OSP.ReadToOtherSystems [tr] The ·*electronic voting system*· shall not obstruct the use of alternative ·*tallying software*· to calculate results.

BWGV-A1 B[3.4g/h]
CoE [26a]
PTB DR[2-5, 2-6]
Paper Funct_7

O.OSP.AccurCalc [di] The ·*tallying software*· shall accurately calculate and display results using the appropriate algorithm based on all ·*e-votes*· stored in the ·*e-ballot box*· and only based on these ·*e-votes*·.

BWGV-A1 B[3.4b/c/d, 3.5a]
CoE [7, 98]
PTB VP[5-1], WS[1-2],
DR[2-2, 2-3, 2-5, 2-6]
Paper Sec_22

O.OSP.Delete [di] The ·*electronic voting machines*· shall provide the functionality to completely delete data from previous ·*elections*·.

Paper Sec_5

5.4.3 Functional Requirements for the Audit System

O.OSP.Audit1 [tr] The ·*audit system*· shall provide the functionality to record, monitor, and verify audit data.

CoE [101]

O.OSP.Audit2 [tr] The ·*audit system*· shall protect the integrity and authenticity of audit records.

CoE [83b, 83c, 109]
PTB CF [2-2, 2-6]

O.OSP.Audit3* [tr] The ·*audit system*· shall have access to a reliable time source.

CoE [83b, 84, 84b]

O.OSP.Audit4* [tr] The ·*audit system*· shall record system configuration and ·*election*· configuration on all ·*electronic voting machines*· at least at the following points

- beginning and end of ·*polling phase*·, as well as
- before and after tallying.

CoE [100, 103, 106]

O.OSP.Audit5* [tr] The ·*audit system*· shall check the ·*e-ballot box*· and the ·*ballot*· content for evidence of tampering.

CoE [107]
PTB CF [4-2]

O.OSP.Audit6 [tr] The ·*audit system*· and its records `should` be tamper-resistant and `shall` be tamper-evident.

CoE [83, 109]
PTB CF [4-4]

O.OSP.Audit7* [tr] For every action performed by ·*poll workers*· the ·*audit system*· `shall` record a timestamp, the nature of the action, and the ID of the particular ·*poll worker*·(where available).

CoE [100]

O.OSP.Audit8 [tr] The ·*audit system*· `shall` record (with timestamps, where appropriate) breakdowns, exceptions, malfunctions, and results of any self-checks.

CoE [100, 103c]
PTB CF[2-5, 4-3]

O.OSP.Audit9 [tr] The ·*audit system*· `shall` implement the access control policy defined by the ·*responsible election authority*·.

CoE [23, 56, 104, 105]
PTB [CF 4-4]

O.OSP.Audit10* [tr] The ·*audit system*· `should` not record any information which might endanger the secrecy of the vote. Where such information is stored it `shall` only be accessible to those with appropriate authority.

CoE [106]

5.5 Assurance Requirements

Assur.1 [all] The ·*responsible election authority*· `shall` define the trust model for their particular ·*election*·.

–

Assur.2 [all] The ·*manufacturer*· `shall` develop the ·*electronic voting machines*· according to software engineering best practice, including use of version control, and bug tracking for all documents and source code.

BWGV-A1 B[2.1a]
PTB CF [1-4, 1-5]

Assur.3* [all] The ·*manufacturer*· `shall` produce the following documents ensuring that they are exhaustive, consistent, unambiguous, appropriate, comprehensible, and concise:

- Complete system specification
- Implemented security functions
- Requirement conformance claim
- Description of each component
- Environmental assumptions
- Testing record
- Development security measures
- User-guide containing
  - normal use instructions for all ·*users*· for all phases
  - appropriate responses to all system messages
- delivery procedure

BWGV [§2(6),§7(1a)]
BWGV-A1 B[1, 4]
PTB PE [3-1, 4-1, 4-2],
VP [5-5],
CF [1-4, 1-6]

Assur.4 [un] The ·*manufacturer*· `shall` build the ·*electronic voting system*· from reliable components.

BWGV-A1 B[2.3]
PTB CF [1-8]

Assur.5* [tr] The ·*manufacturer*· `shall` disclose the documentation from O.OSP.Assur3, executable program, source code, bug tracking, and version control (at least to the ·*testing authority*·).

BWGV-A1 B[1]
CoE [24]
PTB CF [1-1, 3-3]

Assur.6* [all] The ·*manufacturer*· `shall` test the ·*electronic voting machines*·, including functional and usability tests.

CoE [25, 66]
PTB PE [2-4, 4-8, 4-9]

Assur.7 [fr] [un] [non-core] The ·*manufacturer*· `should` involve ·*users*· in the interface development process.

CoE [62]

Assur.8* [all] The ·*testing authority*· `shall` do a risk analysis based on the <u>threat model</u>.

CoE [28]
PTB CF [2-6]

Assur.9* [all] The ·*manufacturer*· shall limit the functionality of the ·*electronic voting machines*· and ·*tallying software*· to that necessary for the ·*election*·.	BWGV-A1 A[f] PTB CF [1-2]
Assur.10* [all] The ·*testing authority*· shall evaluate the ·*electronic voting machines*· against the requirements. Tests shall include penetration, and usability tests.	CoE [25, 28, 72a] PTB PE [2-4, 4-8], CF [1-3, 3-4]
Assur.11* [all] The ·*testing authority*· shall examine the ·*manufacturer's*· documentation from O.OSP.Assur2, executable program, source code, bug tracking, and version control for compliance with requirements and software engineering best practice.	CoE [25, 28, 72a] PTB PE [4-8], CF [1-3, 3-4]
Assur.12* [all] The ·*testing authority*· shall examine the delivery procedures for the ·*electronic voting machines*·, the identified development security measures, and the applied software engineering approach.	CoE [28]

5.6 Additional Requirements

5.6.1 Usability Requirements

Usab.1 [un] All user interfaces shall be user-friendly.	BWGV-A1 B[3.1d] CoE [1b, 65] PTB PE [1-2, 3-1], DR [1-5, 2-4], VP [1-3, 1-7, 3-12]
Usab.2 [un] [fr] All system messages provided by all user interfaces shall be understandable.	BWGV-A1 B[3.7] CoE [1a]
Usab.3 [un] The ·*vote-casting interface*· shall make provision for ·*voters*· with disabilities.	CoE [3, 61, 63]
Usab.4* [eq] The ·*vote-casting interface*· shall clearly indicate to the ·*voter*· whether the ·*electronic voting machine*· is in the ·*active state*·.	–

Usab.5* [tr] The ·*vote-casting interface*· `shall` provide immediate feedback to the ·*voter*· regarding the status of his ·*vote*·.	BWGV-A1 B[3.6g] CoE [14] PTB VP [3-18, 3-19]
Usab.6* [fr] The ·*vote-casting interface*· `shall` protect the ·*voter*· from accidentally ·*casting*· his ·*vote*·	BWGV-A1 B[3.6f] CoE [10] PTB VP [3-9, 3-14]
Usab.7* [all] The ·*poll worker interface*· `shall` protect the ·*poll workers*· from taking any action accidentally.	–
Usab.8 [un] [tr] All used methods `shall` be efficient, thus, the ·*voting process*· does not take more time as necessary.	PTB CF [3-5]

5.6.2 Operational Requirements

Op.1* [all] The ·*responsible election authority*· `shall` educate ·*poll workers*· in the use of the ·*electronic voting machines*· and `shall` ensure that information provided to them is understandable.	BWGV [§7(3)] CoE [1a, 20]
Op.2* [di] The ·*responsible election authority*· `shall` ensure that ·*election data*· is stored with its authentication codes from ·*electronic voting machines*· (and, where applicable, from the ·*tallying software*·) for the prescribed ·*archiving phase*·.	CoE [75c, 99] PTB WS [1-1, 2-2, 2-5]
Op.3* [fr] [un] The ·*responsible election authority*· `shall` educate ·*voters*· in the use of the ·*electronic voting machines*· and `shall` ensure that the information provided to them is understandable.	BWGV [§8(1c)] CoE [1a, 20, 22, 38, 46, 61, 62] PTB PE [2-6, 4-2], VP [3-6]

Op.4* [all] The ·*responsible election authority*· `shall` develop procedures covering all stages of the ·*election*· including

- secure storage of ·*electronic voting machines*· at all times
- logistics (transport of ·*electronic voting machines*·, spare ·*electronic voting machines*·, accessories, etc.)
- configuration of ·*electronic voting machines*· (including ·*ballot*· details and order on ·*electronic voting machines*· and ·*tallying software*·)
- checking ·*electronic voting machines*· (including configuration and empty ·*e-ballot box*·)
- response to any kind of exceptions, malfunctions and breakdowns
- recording of ·*poll worker*· activities, ·*electronic voting machine*· state changes, system resumings, etc.
- ensuring that ·*electronic voting machines*· are in the appropriate state at every stage in the ·*election phase*·.
- closing the ·*poll(s)*·, including disabling ·*electronic voting machines*·
- tallying and re-tallying
- comparing number of ·*votes*· recorded with number of ·*electors*·
- ·*archiving phase*· including data deletion at the end

BWGV [§7(2), §8(1,2), §10(1), §11(5), §12, §13, §14(1,3,5), §15 (1,3), §16, §17(3)]
BWGV-A1 B[2.6]
CoE [28, 29, 31, 51, 52b, 69b, 73, 74, 75, 75a, 77, 79]
PTB PE [2-1, 2-2, 2-3, 2-5, 4-3, 4-6, 4-11],
VP [1-4, 3-3, 4-5],
DR [1-1, 1-3, 1-4],
WS [1-2, 2-1, 2-4,2-6],
CF [1-11]

Op.5 [tr] The ·*responsible election authority*· `shall` develop a contingency plan describing appropriate responses to at least the following circumstances:

- Results produced by recount or alternative ·*tallying software*· do not agree with original result
- Number of ·*votes*· recorded does not match number of ·*electors*·
- Any kind of exceptions, malfunctions, and breakdowns
- Case where ·*voter*· leaves a ·*electronic voting machine*· in ·*active state*·

BWGV [§11(4), §14(5), §15(2)]
CoE [28, 70a, 71a, 75b]
PTB PE [4-11, 4-12],
VP [4-7, 5-5]

Op.6** [all] The ·*responsible election authority*· `shall` define all ·*responsible election authority*· variables, prescribe the certification process (including decertification and recertification) and appoint the ·*testing authority*· and the ·*certification authority*·.

BWGV [§1, §2(3,4), §3, §4]
CoE [85, 111]

Op.7* [all] The ·*responsible election authority*· `shall` define (for all ·*election*· phases)

- timetables,
- access control policy (including separation of duties and minimum team size) inclusive audit data and system related access control,
- administration activities,
- ·*user*· roles,
- key management policy,
- incident levels, and
- reporting procedures.

BWGV [§10(2)]
CoE [23, 28, 32, 33, 36, 56 74, 76, 80, 81, 104]
PTB PE [4-3, 4-4], WS [2-3], CF [3-6]

Op.8* [tr] [non-core] Before the ·*election*· the ·*responsible election authority*· `shall` publicly disclose all technical information about the ·*electronic voting machines*· (including design, configuration, version numbers for all software, etc.).

Appl. Note: Exceptions are only acceptable where it can be shown that such a disclosure would either endanger the security of the ·*electronic voting system*· or genuinely endanger the intellectual property of the ·*manufacturer*·.

BWGV [§6(b)]
CoE [20, 21, 24, 28, 69a]

Op.9* [tr] [non-core] The ·*responsible election authority*· `should` arrange alternative ·*tallying software*· to check results.

CoE [28]

Op.10* [un] [non-core] The ·*responsible election authority*· `shall` clearly indicate whether the ·*electronic voting machines*· are being used in a real ·*election*·.

CoE [50]

Op.11* [fr] [non-core] The ·*responsible election authority*· `should` ensure that all ·*electronic voting machines*· display the ·*ballot*· in a uniform way.	PTB VP [3-4]
Op.12* [tr] [non-core] Before the ·*election*·, the ·*responsible election authority*· `shall` inform ·*voters*· about polling stations where the ·*electronic voting system*· will be used.	BWGV [§6(a)]
Op.13 [all] The ·*poll workers*· `shall` follow the procedures described by the ·*responsible election authority*·.	BWGV [§7(1), §10(2b)] CoE [71b, 73, 77]
Op.14* [all] The ·*poll workers*· `shall` respond to system messages in accordance with the user-guide.	PTB PE [4-11]

5.7 Summary

This chapter defines the exact target of evaluation, stand-alone direct recording electronic voting machines used in polling stations, and itemises these requirements for this type of electronic voting systems. This list contains 59 system requirements (while these are divided in 12 security requirements, 38 functional requirements, and eight usability requirements), 12 assurance, and 14 operational requirements. According to Sect. 4.4, all requirements are labelled by the election principle(s) from which they are deduced. In addition, the defined requirements refer to corresponding requirements in [37], [143], and [62]. Requirements from these documents that are not referred to the requirements for remote electronic voting systems in Chap. 6 or treated in appendix D[4].

Section 5.1 clarifies the relationship between the requirements in this chapter and those provided in [156]. In order to be able to refer to this paper an additional notation is introduced. Then, Sect. 5.2 describes the exact target of evaluation, in particular that the considered systems do not provide voter registration, voter identification, voter authentication, or archiving functionality; that is, only the functionality for the polling phase and the tallying phase are considered.

The 12 security requirements in Sect. 5.3 are deduced from corresponding threats which are also specified (including the type of attack and the

[4] Here, it is explained why particular requirements are not referred to the requirements presented in this book.

motivation). These requirements are divided into those for the polling phase and those for the tallying phase. The functional requirements, in Sect. 5.4, are composed of 26 requirements for the polling phase, three requirements for the tallying phase, and 10 requirements for the audit system. While the security requirements are deduced from threats the functional requirements are related to policies. However, these policies are not specified due to space reasons. Assurance requirements, in Sect. 5.5, address either the tasks of the manufacturer (and thus the development process), the testing authority (how to evaluate the system), or the responsible election authority. In addition, Sect. 5.6 presents the list of usability and organisational requirements, while the last category addresses only responsible election authority tasks as well as documents and procedures to define.

The requirements specified in this chapter serve as basis for the requirement definition for remote electronic voting systems.

6

Requirements for Remote Electronic Voting

The previous chapter specifies the requirements for stand-alone direct recording electronic voting machines. Based on this list and as a result of the development procedures from Sect. 4.1 this chapter provides the requirements for remote electronic voting systems. This standardised, consistent, and exhaustive list of requirements respects the glossary and syntax introduced in Sect. C and 4.4 respectively. The partition of this chapter is equal to the one from chapter 5: Before providing the list of requirements, the chapter-specific notation is explained (meaning those notations not used in the requirements for stand-alone direct recording electronic voting machines). Then, the exact target of evaluation is defined. The partition for the requirements is also taken over from the previous section: First, the security and functional requirements are defined and then, the assurance, usability, and operational requirements.

6.1 Citation and Additional Notations

Phase 3 in the requirement development process for remote electronic voting systems (the first draft of requirements), is based on the requirements for stand-alone direct recording electronic voting machines from chapter 5[1]. Additionally, the requirements listed in the GI/BSI/DFKI Protection Profile [161][2] are considered as part of phase 4 of the development process (improvement based on existing literature).

Notation. As stated in Sect. 4.4, only those requirements from existing catalogues which do not already appear in chapter 5 (because they only address

[1] Note, two requirements categorised as security requirements in Chap. 5 are shifted to functional ones in this chapter, namely: O.T.AvailInfo and O.T.SepDuty. Vise versa, O.OSP.DeleteRecord is shifted from the list of functional requirements to the list of security requirements.

[2] The GI/BSI/DFKI Protection Profile is introduced and discussed in Sect. 8.2.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 93–113, 2009.

remote electronic voting) are referred in this chapter. To indicate the relationship between the requirements from the GI/BSI/DFKI Protection Profile [161] the security and functional requirements are labelled with "BSI name". Correspondingly, "Chap5 name" refers to requirements in the previous chapter for stand-alone direct recording electronic voting machines.

The security and functional requirements are distinguished between those involving the polling phase and those involving 'only' the phase after the polling. Moreover, the requirements are further distinguished according to the component that is addressed: the remote electronic voting system in general, the voting server, the tallying software the client-side voting software, or the audit system.

6.2 Target of Evaluation

The electronic voting system focused on in this chapter is called remote electronic voting system as defined in Sect. 2.1. The idea is to use such a system in parallel to postal voting, that is, every voter who is eligible to cast a postal vote can now choose between the postal or the electronic channel.

A remote electronic voting system can provide more or less functionality. Systems addressed here do not cover all possible implementation techniques and not all election phases. This section describes the target of evaluation.

Covered Functionality. The following operations for the poll workers are addressed:

- Identification and authentication
- Starting the polling phase
- Making a selection on the ballot
- Resuming the polling phase after any kind of exceptions, malfunction, or breakdown
- Checking the system state
- Closing the polling phase
- Starting the tallying phase

In addition, the following operations for the voters are addressed:

- Identification and authentication
- Changing a selection before casting
- Inducing vote casting
- Casting the vote
- Cancelling the voting process

Functionality Not Covered. The election setup and archiving phase are not addressed in the later security and functional requirements definition (analogously to the target of evaluation description in Sect. 5.2). Additionally, the following functionalities which might be implemented in some remote electronic voting systems are out of the scope of the following examinations:

- Running two or more polls in parallel[3]
- Keeping confidential who cast a vote
- Voter or universal verification procedures
- Resistant against disputations
- Changing the electoral register during the polling phase
- Statistical data collection
- Vote updating

If these are implemented, the requirement set needs to be adapted. For instance, to enable the application of more than one poll in parallel demands at least the following additional requirement: the *·remote electronic voting system·* `shall` prevent anyone from linking different *·e-votes·* from the same *·voter·* to one another when polls are run. This is caused by the following threat: an inside intruder with access to *·e-votes·* after the *·polling phase·* discovers some aspect of *·voters'·* identities by examining *·votes·* that were cast together. For instance, non-citizen residents may have limited voting rights. An intruder could determine which votes came from a particular community.

Covered Techniques. Section 2.4 illustrate that there is no best solution for the voter authentication, to ensure the secrecy of the vote, and to implement the client-side voting software. Thus, it is tried to allow all of these techniques for the target of evaluation. Any of the authorisation techniques from Sect. 2.4, are possible implementations. The supported techniques to ensure the secrecy of the vote according to Sect. 2.4 are 'anonymisation in the polling phase and the tallying phase'. The possible voting client techniques, according to Sect. 2.4, are the 'thin and the fat client' approach: that is, computations on the client-side are required.

Techniques Not Covered. Systems implementing "anonymisation in the election setup phase" as a technique to ensure the secrecy of the vote are not covered. For those systems, some of the requirements can be removed because they are passed already by design decisions. However, corresponding requirements need to be defined for the election setup phase. The application of Web browser solutions is only possible if some of the requirements are defined as assumptions about the environment. This is further discussed in Sect. 8.2.3.

Scope. According to the description in Sect. 2.4, a remote electronic voting system includes the voting server (hardware and software), the client-side voting software, the vote-casting device, and the tallying software.

'One' Voting Server. This section only considers one voting server to be generic and to match as many different remote electronic voting systems as possible. However, existing remote electronic voting systems usually distinguish between two or even more voting servers: some are generic and provide n voting servers depending on the configuration. Note, in the case of more

[3] Therefore, the security requirements O.T.LinkInParalElec from Sect. 5.3.2 is not further discussed for remote electronic voting systems.

than one voting server where the voting servers communicate with each other, additional requirements for this communication must be added.

Assumptions. Remote electronic voting belongs to the voting forms where the voter casts his vote in an unprotected environment. As it is proposed to apply remote electronic voting only in parallel to postal voting, problems and corresponding requirements (like coercion resistance) caused by unprotected environments are not addressed in the requirement definition as these problems are already accepted within postal voting (assumption A.ProtectedEnvironment). In addition, for the further considerations, it is assumed that if the remote electronic voting system is set up correctly, it contains the proper electoral register and candidate list as well as the proper definition of valid and invalid votes (assumptions A.ProperConfig). However, there is a requirement (O.OSP.SelfCheck) demanding that poll workers have the possibility of checking the configuration before starting the polling phase. Moreover, it is assumed that (if necessary) the distribution of identification and authentication tokens succeeded and, thus, only but all voters have an identification and authentication token (assumption A.AuthToken).

6.3 Security Requirements

6.3.1 Security Requirements for the Polling Phase

(a) Security Requirements for the Remote Electronic Voting System

T.IneligVoter: An *·ineligible voter·* *·casts·* a *·vote·* in order to affect the *·election result·*.	BSI T.UnauthorisedVoter
O.T.IneligVoter [eq] The *·remote electronic voting system·* `shall` unambiguously identify and authenticate the *·voter·* before storing his *·vote·* in the *·e-ballot box·*.	CoE [82, 94a] PTB VP [1-1] BSI O.UnauthorisedVoter
T.OneVoterOneVote: A malicious *·elector·* *·casts·* a second *·vote·* in order to affect the *·election result·*.	BSI T.UnauthorisedVoter Chap5 T.UnauthVotesA
O.T.OneVoterOneVote [eq] The *·remote electronic voting system·* `shall` store in the *·e-ballot box·* only one *·vote·* per *·voter·*; it `shall` store the first received *·vote·* per *·voter·*.	CoE [5b] BSI O.UnauthorisedVoter Chap5 O.T.UnauthVotes
T.UnauthVotes: An inside intruder adds *·e-votes·* to the *·e-ballot box·* at the *·voting server·* in order to affect the *·election result·*.	BSI A.ElectionOfficers Chap5 T.UnauthVotesB

O.T.UnauthVotes [di] The ·*remote electronic voting system*· `shall` store in the ·*e-ballot box*· only ·*e-votes*· cast from ·*eligible voters*·. Any other access to the ·*e-ballot box*· `shall` be denied.	Chap5 O.T.UnauthVotes
T.PersonalDataNet: An outside intruder sniffs the network in order to collect personal data from ·*voters*·.	–
O.T.PersonalDataNet [dp] The ·*remote electronic voting system*· `shall` ensure the data protection law with respect to the transmission of any personal data.	BSI O.SecretMessage
T.SecretAuthNet: An outside intruder sniffs the network to get ·*authentication information*· and to use this to ·*cast*· a ·*vote*· on behalf of a ·*voter*· in order to affect the ·*election result*·.	–
O.T.SecretAuthNet [un] The ·*remote electronic voting system*· `shall` protect the confidentiality of the transmitted ·*authentication information*·.	BSI O.SecretMessage
T.IntResultNet: The outside intruder sniffs the network in order to compute intermediate results.	BSI T.SecretMessage
O.T.IntResultNet [fr] The ·*remote electronic voting system*· `shall` ensure the confidentiality of the transmitted ·*e-votes*· during the ·*polling phase*·.	BSI O.SecretMessage
T.DeleteMsgNet: Unnoticed, the outside intruder deletes messages in the network to exclude ·*voters*· from the ·*election*· in order to affect the ·*election result*· or in order to confuse ·*voters*·.	BSI T.IntegrityMessage
O.T.DeleteMsgNet [un] [tr] The ·*remote electronic voting system*· `shall` ensure that protocol messages cannot be deleted undetected.	PTB VP[4-6a], DR[1-2b] BSI T.IntegrityMessage
T.AlterMsgNet: An outside intruder unnoticed replays old protocol messages, sends new ones, or alters messages in order to affect the ·*election result*·.	BSI T.IntegrityMessage

O.T.AlterMsgNet [all] The ·*remote electronic voting system*· `shall` verify the freshness, authenticity, integrity, and format correctness of all messages before processing them.

BSI O.IntegrityMessage

T.DeleteRecord: An outside intruder uses the ·*voter's*· ·*vote-casting device*· after the ·*voter*· ·*cast*· his ·*vote*· in order to compromise the secrecy of the vote.

BSI A.Buffer

O.T.DeleteRecord [se] The ·*remote electronic voting system*· `shall` delete any records related to the ·*voter's*· ·*voting process*· from the ·*vote-casting device*· when finishing the ·*voting process*·.

Chap5 O.OSP.DeleteRecord

T.ElecSecrecyNet: An outside intruder sniffs the network in order to compromise the secrecy of the vote.

BSI T.SecretMessage

O.T.ElecSecrecyNet [fr] The ·*remote electronic voting system*· `shall` not provide any information in the transmitted protocol messages, which allows to construct the link between a particular ·*voter*· and his ·*vote*·. The ·*remote electronic voting system*· `shall` ensure that neither the ·*vote*· itself nor the number of chosen ·*voting options*· (including an empty ·*ballot*·), nor a ·*spoilt*· ·*vote*· (for example, by using the length of the protocol messages) can be linked to a particular ·*voter*·. In addition, it `shall` be ensured that the sequence of messages does not reveal the link.

BSI O.SecrecyOfVoting

T.ProofGenA: A malicious ·*elector*· uses all information either sent to, displayed on, and/or sent from his ·*vote-casting device*· to construct a proof in order to sell his ·*vote*·.

BSI T.Proof

T.ProofGenB: A malicious ·*elector*· uses all information from T.ProofGenA and intermediate results calculated on his ·*vote-casting device*· to construct a proof in order to sell his ·*vote*·.

BSI T.Proof

O.T.ProofGen [se] The ·*remote electronic voting system*· `shall` ensure that ·*voters*· are not able to construct a receipt proving their ·*vote*·. Neither information sent to, displayed on, sent from, nor intermediate results calculated on his ·*vote-casting device*· or protocol messages sequences `shall` serve as proof.

CoE [93b]
BSI O.Proof
Chap5 O.OSP.DeleteRecord

(b) Security Requirements for the Voting Server

T.WrongSW: An outside intruder disseminates manipulated ·*client-side voting software*· in order to reach any of his goals.	–
O.T.WrongSW [all] The ·*voting server*· `shall` communicate only with the authentic and unaltered ·*client-side voting software*·.	–
T.TamperServerA: An outside intruder gets access to the ·*voting server*· over the network and tampers with it in arbitrary ways in order to reach any of his goals.	BSI A.ElectionServer
T.TamperServerB: An inside intruder tampers with the ·*voting server*· in arbitrary ways in order to reach any of his goals.	BSI A.ElecttionOfficers Chap5 T.Tamper
O.T.TamperServer [all] The ·*voting server*· `should` be tamper-resistant. The ·*voting server*· `shall` be tamper-evident.	PTB VP[2-3] Chap5 O.T.Tamper
T.AC: An outside intruder gets access to the ·*voting server*· without knowing or having the access tokens to tamper with the ·*voting server*· in order to reach any of his goals.	BSI P.AuthElectionOfficers A.ServerRoom Chap5 T.AC
O.T.AC [all] The ·*voting server*· `shall` implement an access control policy for the ·*poll worker interface*· which • restricts all activities to particular ·*user*·-roles and • requires physical presence.	BSI O.AuthElectionOfficers Chap5 O.T.AC
T.ElectionSecrecyA: An outside intruder accesses the ·*election data*· after the ·*polling phase*· in order to compromise the secrecy of the vote.	BSI T.ArchivingSecrecyOfV. Chap5 T.ElectionSecrecy
T.ElectionSecrecyB: An inside intruder gets access to the ·*voting server*· and uses stored information in order to compromise the secrecy of the vote.	BSI A.ElectionOfficers Chap5 T.ElectionSecrecy

O.T.ElectionSecrecy [se] The ·*voting server*· should not store any information which could link the ·*voter*· with his ·*vote*· after the completion of the ·*voting process*·. Where any information which could link the ·*voter*· to his ·*vote*· is stored on the ·*voting server*·, it shall only be accessible to those with appropriate authority.

BSI O.ArchivingSecrecyOfV.
Chap5 O.T.ElectionSecrecy

(c) Security Requirements on the Client-Side

T.TamperClient: An outside intruder runs malware on the ·*vote-casting device*·, which either reads the ·*vote*· (in order to compromise the secrecy of the vote), alters the ·*vote*·, or reads the authentication information to ·*cast*· a ·*vote*· or to bar the ·*voter*· from ·*casting a vote*· (in order to affect the ·*election result*·).

BSI A.VoteCastingDevice
Chap5 T.Tamper

O.T.TamperClient [all] The ·*client-side voting software*· shall ensure that its operations and data are unaffected by other applications running on the ·*vote-casting device*·.

Chap5 O.T.Tamper

T.WrongServer: An outside intruder tries to redirect the ·*voter*· to a faked ·*voting server*· in order to reach any of his goals.

BSI T.AuthenticityServer

O.T.WrongServer [all] The ·*client-side voting software*· shall only communicate with the authentic and unaltered ·*voting server*·.

CoE [90b]
BSI O.AuthenticityServer

6.3.2 Security Requirements for the Tallying Phase

T.IntegVotes: An inside intruder tampers with ·*e-votes*· after the ·*polling phase*· and before the ·*tallying phase*· in order to affect the ·*election result*·.

Chap5 T.IntegVotes

O.T.IntegVotes [di] The ·*voting server*· shall protect the integrity and authenticity of ·*e-votes*· after the ·*polling phase*·.

BSI O.ArchivingIntegrity
Chap5 O.T.IntegVotes

O.T.AuthCheckCount [di] The ·*tallying software*· shall verify the integrity and authenticity of ·*e-votes*·.

Chap5 O.T.AuthCheckC.

T.IntegElecData: An inside intruder tampers with ·*election data*· after the ·*tallying phase*· in order to affect the ·*election result*· in case of recounts.

BSI T.ArchivingIntegrity
Chap5 T.IntegElecData

O.T.IntegElecData [di] The ·*tallying software*· `shall` protect the integrity and authenticity of ·*election data*· as soon as the tallying is completed.

BSI O.ArchivingIntegrity
Chap5 O.T.IntegElecData

T.AffectCounting: An inside intruder installs malware on the machine running the ·*tallying software*· in order to affect the ·*election result*·.

BSI A.ElectionOfficers
Chap5 T.AffectCounting

O.T.AffectCounting [di] The ·*tallying software*· `shall` ensure that its operations and data are unaffected by other applications.

Chap5 O.T.AffectCounting

6.4 Functional Requirements

6.4.1 Functional Requirements for the Polling Phase

(a) Functional Requirements for the Remote Electronic Voting System

O.OSP.VoteRight [un] [di] The ·*remote electronic voting system*· `shall` ensure that no ·*voter*· looses his voting right without having ·*cast a vote*·.

BSI P/O.OneVoterOneVote

O.OSP.NoInteract [un] The ·*remote electronic voting system*· `shall` prevent ·*voter*· interactions in case of exceptions and malfunctions.

Chap5 O.OSP.NoInteract

O.OSP.Confirmation [tr] The ·*remote electronic voting system*· `shall` provide a confirmation to the ·*voter*· regarding the status of his ·*vote*· – at least the information that his ·*e-vote*· has been successfully stored.

Appl. Note: In case the ·*voter*· does not receive the confirmation, he shall get this information as soon as he logs on again.

BSI P/O.Acknowledgement
Chap5 O.OSP.PosFeedback

O.OSP.Feedback [un] The ·*remote electronic voting system*· `shall` provide feedback to the ·*poll workers*· in form of error messages in case of exceptions, malfunctions, and breakdowns. Where a ·*voter*· is in the ·*voting process*· at that time he `shall` also get a feedback.

Chap5 O.OSP.NegFeedback

O.OSP.DataLoss [di] The ·*remote electronic voting system*· shall prevent data loss during normal operations and in case of exceptions, malfunctions, and breakdowns.

PTB VP [2-1b]
BSI A/OE.DataStorage
Chap5 O.OSP.DataLoss

O.OSP.Availability [un] [non-core] The ·*remote electronic voting system*· should be available during the whole ·*polling phase*·.

Appl. Note: The ·*remote electronic voting system*· shall be robust against power outage at the ·*voting server*·, unexpected ·*user*· activity, environmental effects (for instance, mechanical, electromagnetic, and climatic) to the ·*voting server*·, and network problems.

PTB CF [2-1]
BSI A/OE.Availability
Chap5 O.OSP.Avaliability
Chap5 O.OSP.Robust

O.OSP.VoteRightExc [un] [di] The ·*remote electronic voting system*· shall ensure that in case of exceptions, malfunctions, and breakdowns no ·*voter*· looses his right to ·*cast*· a ·*vote*· nor get the possibility to ·*cast*· two ·*votes*·.

Appl. Note: The ·*remote electronic voting system*· shall be capable to determine whether a particular ·*voter*· ·*cast*· a vote and his ·*e-vote*· was successfully stored in case of exceptions, malfunctions, and breakdowns.

PTB VP[1-5]
BSI P/O.OneVoterOneVote
Chap5 O.OSP.LastVote

(b) Functional Requirements for the Voting Server

O.OSP.SepDuty [all] The access control mechanism shall only allow access to the ·*voting server*· if at least two different ·*users*· are logged on.

BSI P/O.AuthElectionO.
Chap5 O.T.SepDuty

O.OSP.Auditing [tr] The ·*voting server*· shall be capable of producing comprehensive audit data.

BSI P/O.Audit
Chap5 O.OSP.Autditng

O.OSP.InfoPW [di] [non-core] The ·*voting server*· shall indicate to the ·*poll worker*·

- the number of ·*votes*· ·*cast*· so far and
- its current state.

Chap5 O.OSP.InfoPW

O.OSP.StoreAllVotes [di] The·*voting server*· shall store in the ·*e-ballot box*· all ·*e-votes*· ·*cast*· by ·*eligible voters*· during the ·*polling phase*·.

Chap5 O.OSP.StoreAllVotes

O.OSP.PWClosePoll [un] The ·*poll worker interface*· `shall` warn the ·*poll workers*· if they try to close the ·*election*· before the final date.

BSI P/O.EndingElection
Chap5 O.OSP.PWClosePoll

O.OSP.AvailInfo [tr] The ·*voting server*· `shall` not provide any information about the ·*voting process*· except the current state and the number of ·*votes*· ·*cast*· so far.

BSI P/O.SecrecyOfVotingElec.
Chap5 O.T.AvailInfo

O.OSP.SelfCheck [all] The ·*voting server*· `should` regularly perform automatic self-checks and report the results to the ·*poll workers*·. The ·*voting server*· `shall` be capable of performing self-checks.

BSI P/O.Failure
Chap5 O.OSP.SelfCheck

O.OSP.ErrorRecovery [di] [un] The ·*voting server*· `shall` run a self-check before a resuming is possible. In case of irreversible problems the ·*voting server*· `shall` prevent a resuming of the ·*polling phase*·.

BSI P.Failure
Chap5 O.OSP.ErrorRecovery

O.OSP.PWInterface [se] [fr] The only functionality provided by the ·*poll worker interface*· is

- identification and authentication,
- starting the ·*polling phase*· which is only possible once,
- resuming the ·*polling phase*· after any kind of exceptions, malfunctions, and breakdowns according to O.OSP.ErrorRecovery,
- closing the ·*polling phase*· after which the actions ‘starting’ and ‘resuming’ are disabled,
- starting the ·*tallying phase*· only after having closed the ·*polling phase*·,
- performing self-checks,
- checking that the ·*voting server*· has been set up correctly (for example, order of ·*voting options*· and empty ·*e-ballot box*·),
- checking the current state according to O.OSP.InfoPW, and
- reading the audit trails.

Appl. Note: The ·*voting server*· shall not provide any functionality to reach any of the intruder’s goals described in Sect. 4.3.

BSI P/O.EndOfElection
P/O.IntegrityElectionOfficers
P/O.IntermediateResult
P/O.Failure
P/O.Audit
P/O.StartTallying
Chap5 O.OSP.PWInterface
O.OSP.PWCheck

O.OSP.SecrecyAfterBreakd [se] In case of exceptions, malfunctions, and breakdowns, the ·*voting server*· `shall` not reveal the link from the last ·*voter*· to his ·*selections*· or ·*vote*·.

Chap5 O.T.SecrecyAfterBr.

O.OSP.ClosePoll [un] [non-core] The acceptance of ·*e-votes*· into the ·*e-ballot box*· `should` remain open for a sufficient phase of time to allow for any delay of data transport.

PTB DR[1-2a]
CoE [96b]

O.OSP.AdequNoVotes [un] [non-core] The ·*voting server*· `shall` be capable of recording an adequate number of ·*votes*·.

Chap5 O.OSP.AdequNoVotes

O.OSP.AdequNoBallotOpt [fr] [non-core] The ·*voting server*· `shall` support an adequate number of ·*voting options*·.

Chap5 O.OSP.AdequNoBall.

(c) Functional Requirements for the Client-Side Voting Software

O.OSP.Interface [fr] The ·*client-side voting software*· `shall` provide the following functionality for the ·*voter*·:

- Identification and authentication
- Make a choice on the ·*ballot*·
- Change ·*selections*· before ·*casting a vote*·
- Initialise vote casting
- ·*Vote casting*·
- Cancel his ·*voting process*· at any time

BSI P/O.Abort,
P/O.Correction,
P/O.OverhasteProtection
Chap5 O.OSP.V-Interface

O.OSP.AccurDisp [fr] The ·*voting server*· `shall` accurately display the authentic and unaltered ·*ballot*·.

Chap5 O.OSP.AccurDisp

O.OSP.Transmission [un] The ·*client-side voting software*· `shall` immediately transmit the ·*e-votes*· to the ·*voting server*·, whenever a ·*voter*· has ·*cast*· his ·*vote*·.

–

O.OSP.Spoil [fr] [non-core] The ·*client-side voting software*· `should` provide the functionality for the ·*voter*· to ·*spoil*· his ·*vote*·.

Chap5 O.OSP.Spoil

O.OSP.SpoilWarning [fr] [non-core] The ·*client-side voting software*· `should` warn the ·*voter*· when he tries to ·*spoil*· his ·*vote*· in one or more ·*polls*·.

Chap5 O.OSP.SpoilWarning

O.OSP.EqualPres [fr] The ·*client-side voting software*· `shall` ensure equality and accuracy of presentation of ·*voting options*· on any ·*vote-casting device*·.

Appl. Note: The ·*remote electronic voting system*· shall avoid the display of other influencing messages.

Chap5 O.OSP.EqualPres

O.OSP.AccurRep [fr] The ·*client-side voting software*· `shall` ensure that the ·*voter's*· ·*selections*· are accurately represented in the ·*e-vote*·.

Chap5 O.OSP.AccurRep

O.OSP.CompatClient [fr] [non-core] The ·*client-side voting software*· `should` be compatible with any ·*vote-casting device*· and with devices used by people with disabilities where appropriate.

Chap5 O.OSP.CompatClient

6.4.2 Functional Requirements for the Tallying Phase

O.OSP.ReadToOtherSystems [tr] The ·*remote electronic voting system*· `shall` provide the functionality to upload ·*e-votes*· into any ·*tallying software*·.

Chap5 O.OSP.ReadToO.

O.OSP.DeleteData [di] The ·*voting server*· `shall` provide the functionality to completely delete all data from previous ·*elections*·.

Chap5 O.OSP.Delete

O.OSP.AccurCalc [di] The ·*tallying software*· `shall` accurately calculate results using the appropriate algorithm based on all (authorised) ·*e-votes*· stored in the ·*e-ballot box*· and only based on these ·*e-votes*·.	BSI P/O.Tallying Chap5 O.OSP.AccurCalc

6.4.3 Functional Requirements for the Audit System

O.OSP.Audit1 [tr] The ·*audit system*· `shall` provide the functionality to record, monitor, and verify audit data.	Chap5 OSP.Audit.1
O.OSP.Audit2 [tr] The ·*audit system*· `shall` protect the integrity and authenticity of audit records.	Chap5 O.OSP.Audit.2
O.OSP.Audit3 [tr] The ·*audit system*· `shall` have access to a reliable time source.	Chap5 O.OSP.Audit.3 BSI OE.SystemTime
O.OSP.Audit4 [tr] The ·*audit system*· `shall` record system configuration (including software version numbers) and ·*election*· configuration (including ·*voting option*· information) on the ·*voting server*· at least at the following points • beginning and end of ·*polling phase*·, as well as • before and after tallying.	Chap5 O.OSP.Audit.4
O.OSP.Audit5 [tr] The ·*audit system*· `shall` check the ·*e-ballot box*·, the ·*ballot*· content, and the ·*authentication data*· for evidence of tampering.	Chap5 O.OSP.Audit.5
O.OSP.Audit6 [tr] The ·*audit system*· and its records `should` be tamper-resistant and `shall` be tamper-evident.	Chap5 O.OSP.Audit.6 BSI OE.AuditTrailProt.

O.OSP.Audit7 [tr] For every action performed by ·*poll workers*· the ·*audit system*· `shall` record

- a timestamp,
- the nature of the action, and
- the ID of the particular ·*poll worker*·(where available).

Chap5 O.OSP.Audit.7

O.OSP.Audit8 [tr] The ·*audit system*· `shall` record (with timestamps, where appropriate)

- breakdowns,
- exceptions,
- malfunctions, and
- results of any self-checks.

Chap5 O.OSP.Audit.8

O.OSP.Audit9 [tr] The ·*audit system*· `shall` implement the access control policy defined by the ·*responsible election authority*·.

Chap5 O.OSP.Audit.9

O.OSP.Audit10 [tr] The ·*audit system*· `should` not record any information which might endanger the secrecy of the vote. Where such information is stored it `shall` only be accessible to those with appropriate authority.

Chap5 O.OSP.Audit.10

O.OSP.Audit11 [dp] The ·*audit system*· `shall` ensure the data protection law.

CoE [110]

6.5 Assurance Requirements

Some of the assurance requirements are additionally labelled with small letters. These are used in Sect. 7.3 to refer to parts of a particular requirement.

Assur.1 [all] The ·*responsible election authority*· `shall` define the trust model for their particular ·*election*·.

Chap5 Assur.1

Assur.2 [un] The ·*manufacturer*· `shall` develop the ·*electronic voting system*· (a) according to software engineering best practice, including use of (b) version control, and (c) bug tracking for all documents and source code.

Chap5 Assur.2

Assur.3 [all] The ·*manufacturer*· `shall` produce the following documents ensuring that they are exhaustive, consistent, unambiguous, appropriate, comprehensible, and concise: Chap5 Assur.3

(a) Complete system specification
(b) Implemented security functions
(c) Requirement conformance claim
(d) Description of each component
(e) Environmental assumptions
(f) Testing record
(g) Development security measures
(h) User-guide containing
- normal use instructions for all ·*users*· for all phases
- appropriate responses to all system messages

(i) delivery procedure

Assur.4 [un] The ·*manufacturer*· `shall` build the ·*electronic voting system*· from reliable components. Chap5 Assur.4

Assur.5 [tr] The ·*manufacturer*· `shall` disclose (a) the documentation from Assur.2, (b) executable program, (c) source code, (d) bug tracking, and (e) version control (at least to the ·*testing authority*·). Chap5 Assur.5

Assur.6 [all] The ·*manufacturer*· `shall` test the ·*electronic voting system*·, including functional and usability tests. Chap5 Assur.6

Assur.7 [fr] [un] [non-core] The ·*manufacturer*· `should` involve ·*users*· in the interface development process. Chap5 Assur.7

Assur.8 [all] The ·*testing authority*· `shall` do a risk analysis based on the threat model. Chap5 Assur.8

Assur.9 [all] The ·*manufacturer*· shall limit the functionality of the ·*electronic voting system*· and ·*tallying software*· to that necessary for the ·*election*·.	Chap5 Assur.9
Assur.10 [all] The ·*testing authority*· shall evaluate the ·*electronic voting machines*· against the requirements. Tests shall include penetration, and usability tests.	Chap5 Assur.10
Assur.11 [all] The ·*testing authority*· shall examine the ·*manufacturer's*· (a) documentation from Assur.2, (b) executable program, (c) source code, (d) bug tracking, and (e) version control for compliance with requirements and software engineering best practice.	Chap5 Assur.11
Assur.12 [all] The ·*testing authority*· shall examine (a) the delivery procedures for the ·*electronic voting system*·, (b) the identified development security measures, and (c) the applied software engineering approach.	Chap5 Assur.12

6.6 Additional Requirements

6.6.1 Usability Requirements

Usab.1 [un] All user interfaces shall be user-friendly.	Chap5 Usab.1
Usab.2 [un] [fr] All system messages provided by all user interfaces shall be understandable.	Chap5 Usab.2
Usab.3 [un] The ·*vote-casting interface*· shall make provision for ·*voters*· with disabilities.	Chap5 Usab.3
Usab.4 [tr] The ·*vote-casting interface*· shall provide immediate feedback to the ·*voter*· regarding the status of his ·*vote*·.	Chap5 Usab.5

Usab.5 [fr] The ·*vote-casting interface*· `shall` protect the ·*voter*· from accidentally ·*casting*· his ·*vote*· Chap5 Usab.6

Usab.6 [all] The ·*poll worker interface*· `shall` protect the ·*poll workers*· from taking any action accidentally. Chap5 Usab.7

Usab.7 [un] [tr] All used methods `shall` be efficient, thus, the ·*voting process*· does not take more time as necessary. Chap5 Usab.8

Usab.8 [un] The ·*client-side voting software*· `shall` be easy to install on the ·*vote-casting device*·. Chap5 Usab.9

6.6.2 Operational Requirements

Op.1 [tr] The ·*responsible election authority*· `shall` develop a contingency plan describing appropriate responses to at least the following circumstances: Chap5 Op.5

- results produced by recount or alternative ·*tallying software*· do not agree with original result
- number of ·*votes*· recorded does not match number of ·*electors*·
- any kind of exceptions, malfunctions, and breakdowns

Op.2 [all] The ·*responsible election authority*· `shall` define (for all ·*election*· phases): Chap5 Op.7

- timetables
- access control policy (including separation of duties and minimum team size) inclusive audit data and system related access control
- administration activities
- ·*user*· roles
- key management policy
- incident levels
- reporting procedures

Op.3 [un] The ·*responsible election authority*· `shall` provide additional channels to ·*cast*· the ·*vote*· other than the remote electronic voting one.

CoE [4]

Op.4 [all] The ·*responsible election authority*· `shall` develop procedures covering all stages of the ·*election*·, including

- secure ·*voting server*· storage at all times
- ·*voting server*· configuration (including ·*ballot*· details, order on ·*voting server*·, and ·*tallying software*·)
- checking ·*voting server*· (including configuration and empty ·*e-ballot box*·)
- response to any kind of exceptions, malfunctions, and breakdowns
- recording of ·*poll worker*· activities, ·*voting server*· state changes, system resuming, etc.
- ensuring that the ·*voting server*· is in the appropriate state at every stage in the ·*election phase*·.
- closing the ·*poll(s)*·, including disabling ·*voting server*·
- tallying and re-tallying
- comparing number of ·*votes*· recorded with number of ·*electors*·
- ·*archiving phase*·, including data deletion at the end
- ·*identification and authentication token*· delivery, their storage and management where necessary

Chap5 Op.4

Op.5 [all] The ·*responsible election authority*· `shall` define all ·*responsible election authority*· variables, prescribe the certification process (including decertification and recertification), appoint the ·*testing authority*·, and the ·*certification authority*·.

Chap5 Op.6

Op.6 [un] The ·*responsible election authority*· `shall` coordinate the different channels, for instance, it `shall` prevent ·*voters*· ·*casting one vote*· per possible channel and `shall` develop a procedure to merge the results from different channels.

CoE [6, 7, 37, 41, 44, 45, 53]
PTB CF[2-4]

Op.7 [tr] [non-core] Before the ·*election*· the ·*responsible election authority*· `shall` publicly disclose all technical information about the ·*electronic voting system*· (including design, configuration, version numbers, etc.).
Remark: Exceptions are only acceptable where it can be shown that such a disclosure would either endanger the security of the ·*electronic voting system*· or genuinely endanger the intellectual property of the ·*manufacturer*·. Chap5 Op.8

Op.8 [all] The ·*responsible election authority*· `shall` educate ·*poll workers*· in the use of the ·*electronic voting system*· and `shall` ensure that information provided to them is understandable. Chap5 Op.1

Op.9 [di] The ·*responsible election authority*· `shall` ensure that ·*election data*· is stored with its authentication codes (and, where applicable, from the ·*tallying software*·) for the prescribed ·*archiving phase*·. Chap5 Op.2

Op.10 [all] The ·*poll workers*· `shall` follow the procedures described by the ·*responsible election authority*·. Chap5 Op.13

Op.11 [all] The ·*poll workers*· `shall` respond to system messages in accordance with the user-guide. Chap5 Op.14

Op.12 [fr] [un] The ·*responsible election authority*· `shall` educate ·*voters*· in the use of the ·*electronic voting system*· and `shall` ensure that the information provided to them is understandable. Chap5 Op.3

Op.13 [tr] [non-core] The ·*responsible election authority*· `should` arrange alternative ·*tallying software*· to check results. Chap5 Op.9

Op.14 [un] [non-core] The ·*responsible election authority*· `shall` clearly indicate whether the ·*electronic voting system*· are being used in a real ·*election*·. Chap5 Op.10

Op.15 [fr] [non-core] The ·*responsible election authority*· `should` ensure that all ·*electronic voting system*· display the ·*ballot*· in a uniform way. Chap5 Op.11

6.7 Summary

This chapter defines the exact type of considered remote electronic voting systems and itemises all requirements for this type of electronic voting systems. This list contains 71 system requirements (while these are divided in 21 security requirements, 42 functional requirements, and eight usability requirements), 12 assurance and 15 operational requirements. According to Chap. 4 the requirements are labelled by election principle(s). The requirements refer either direct or indirect to corresponding requirements in in [37], [143], and [62] (indirect by referring to requirements from Chap. 5).

Section 6.1 clarifies the relationship between the requirements in this chapter and those provided in the GI/BSI/DFKI Protection Profile [161]. The notations used to refer to the Protection Profile and to requirements from the previous chapter are introduced. Afterwards, Sect. 6.2 describes the exact target of evaluation: the considered functionality for voters and poll workers is defined and it is stated that systems implementing "anonymisation in the election setup phase" as a technique to ensure the secrecy of the vote are not covered as well as systems using the Web browser approach, while all other approaches discussed in Chap. 2.4 are considered. Moreover, it is decided that the evaluation only covers the functionality for the polling phase and the tallying phase. Besides these functional aspects, it is explained why different possible voting servers are subsumed to one voting server. In addition, the assumptions to the environment are presented (A.ProtectedEnvironment, A.ProperConfig, and A.AuthToken).

The 21 security requirements in Sect. 6.3 are deduced from corresponding threats which are also specified. These requirements are divided into those for the polling phase and those for the tallying phase. The functional requirements in Sect. 6.4 are composed of 28 requirements for the polling phase, three requirements for the tallying phase, and 11 requirements for the audit system. Assurance requirements in Sect. 6.5 address either the tasks of the manufacturer (and thus the development process), the testing authority (how to evaluate the system), or the responsible election authority. In addition, Sect. 6.6 specifies the list of usability and organisational requirements. The last category addresses only responsible election authority tasks and mainly document and procedures to define.

As the focus of this book is on security issues, the security, functional, and assurance requirements are treated as input for the next part – the evaluation part – while the organisational and usability requirements are not further discussed.

Part III

Evaluation

7

Evaluation Methodology

The previous two parts provide the fundamentals and specify a list of requirements for stand-alone direct recording electronic voting machines and a second list for remote electronic voting systems. The "requirement part" overcomes one of the identified vulnerabilities of existing evaluation documents for electronic voting systems by defining standardised, consistent, and exhaustive requirements. In this part the remaining identified vulnerabilities are addressed by providing a standardised evaluation methodology, taking the underlying trust model into account and being flexible with respect to different evaluation depths.

This "evaluation part" only considers remote electronic voting systems, while in the previous part, the requirements for electronic voting machines and for remote electronic voting systems are specified. In addition, the proposed evaluation methodology only addresses the security, functional, and assurance requirements for remote electronic voting systems, while the defined operational and usability requirements are not considered. However, the proposed methodology can easily be adapted and extended for stand-alone direct recording electronic voting machines or any other type of electronic voting system.

After a short discussion of established evaluation methodologies, the most appropriate methodology is explained: The Common Criteria (CC) [35] and the corresponding Common Evaluation Methodology (CEM) [36]. In particular, the application of the Common Criteria for remote electronic voting is discussed. In this context, different trust models for remote electronic voting in terms of the Common Criteria approach are discussed in general and in particular their implications for possible implementations of remote electronic voting systems. There are two chosen examples: The "temporary unlimited secrecy of the vote" and the "trustworthiness of the vote-casting device".

In addition, the requirements from Chap. 6 are translated to the Common Criteria syntax; in particular the assurance requirements to the corresponding Common Criteria evaluation level. In order to be able to choose also high Common Criteria evaluation levels requiring formal methods, a first step to

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 117–147, 2009.

develop a formal IT security model is taken. A subset of requirements from Chap. 6 is specified in such a model.

7.1 Common Criteria Introduction

There exists a multiplicity of standards and methodologies for the evaluation of security critical systems. Examples are the IT Grundschutz manual, ISO/IEC 17799 and BS 7799, ISO TR 13335, ITSEC/Common Criteria, FIPS 140-1/2, Task Force Secure Internet, CobiT, and ISO 9000 (for a comparison see [69] and [14]). It can be differentiated between those that refer to individual products or components within an IT landscape and those that address the interaction of several components in an overall IT system. Further, there are standards concerning technical aspects while others concentrate on non-technical (for instance organisational) aspects. The Common Criteria and the FIPS Criteria are suitable exclusive for the examination of IT products, while the IT Grundschutz manual focuses on the interaction of several components in an overall IT system, including aspects such as configuration, organisation and emergency precaution. Since the evaluation of remote electronic voting systems counts among the category of an IT product, the Common Criteria and the FIPS criteria are short-listed. One of the goals of the 1994 of NIST (National Institute of Standards and Technology) published "Security requirements for Cryptographic Modules: FIPS 140-1-Criteria" is to validate cryptographic modules. However, as remote electronic voting systems contain a lot of more security functions than those related to cryptography, the FIPS criteria are not broad enough for the evaluation of remote electronic voting systems[1]. Thus, the evaluation standard that works best is the Common Criteria evaluation methodology.

History and Background. The Common Criteria (CC) [35] is an international standard (ISO 15408) for computer security. The official name is "The Common Criteria for Information Technology Security Evaluation". It results from a standardisation of national security criteria from different sources, starting with the "Orange Book" of the U.S. DOD 1985. The Common Criteria standard is improved continually. At the moment, the official Common Criteria version is V3.1. Today many nations, such as Germany, France, and the U.K. have introduced the Common Criteria to evaluate and certify IT security products and procedures. In addition, there is a growing list of nations which at least accept the CC certificates, such as Greece and Italy. Therefore, certifying an electronic voting system in Germany also grants acceptance in France, Spain, and many other countries.

The CC's purpose is to allow users (here the responsible election authorities and the voters) to specify their security requirements, to allow developers

[1] It might be recommendable to apply the FIPS criteria for the cryptography used to ensure the secrecy of the vote (or in general in the voting protocol) but this is not further discussed in this book).

Part 1	Concepts and Primitives
Part 2	Classes of Security Functional Requirements FAU, FCO, FCS, FDP, FIA, FMT, FPR, FPT, FRU, FTA, FTP
Part 3	Classes of Security Assurance Requirements ASE, ADV, AGD, ALC, ATE, AVA

Fig. 7.1. Structure of the Common Criteria

to specify the security functions of their products, and to allow evaluators to determine if products actually meet their claims. Independent of these three groups an independent certification authority certifies the related statements. The IT product to be evaluated is called the **target of evaluation**[2] **(TOE)**.

Structure. The Common Criteria contains the following three parts (see Fig. 7.1 for an overview):

- The **Introduction and Common Model** (part 1): "It defines the general concepts and principles of IT security evaluation and presents a general model of evaluation." [35];
- The **Security Functional Requirements (SFR)** (part 2): It "establishes a set of functional components that serve as standard templates upon which to base functional requirements[..]. " [35]; The CC assigns each of these security functional requirement to one of the following classes: security audit (FAU), communication (FCO), cryptographic support (FCS), data protection (FDP), identification and authentication (FIA), security management (FMT), privacy (FPR), protection of the TSF[3] (FPT), resource utilisation (FRU), TOE access (FTA), and trusted path/channels (FTP).
- The **Security Assurance Requirements (SAR)** (part 3): It "establishes a set of assurance components that serve as standard templates upon which to base assurance requirements [..] and presents seven predefined assurance packages which are called the evaluation assurance levels (EALs)." [35]. The security assurance requirements are grouped in classes, while the CC distinguishes here between the following classes: security target evaluation (ASE), development (ADV), guidance documents (AGD), life-cycle support (ALC), tests (ATE), and vulnerability assessment (AVA). The Common Criteria approach identifies separate concepts of assurance (a) in a TOE at the end of the evaluation and (b) of maintenance of that assurance during the operational use.

[2] According to the Common Criteria, TOE means "a set of software, firmware and/or hardware possibly accompanied by guidance." [35]

[3] TSF - 'TOE Security Functionality' according to the Common Criteria definition: "a set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs." [35]

Both the security functional and security assurance requirement components are organised into families and these again into classes. Dependencies between different components are figured out. This makes it easier to provide a consistent set of requirements.

Besides the security functional and security assurance requirement components, the other two important components of the Common Criteria are the security problem definition and the security objectives. These are now further discussed.

Security Problem Definition. The security problem definition defines the addressed axiomatic security problem. The usefulness of an evaluation result strongly depends on the quality of the security problem definition and whether it correctly reflects the situation. However, the CC does not define the deriving process to define the security problem.

The security problem definition contains threats, organisational security policies[4] (OSPs) and assumptions "that must be countered, enforced and upheld by the TOE and its operational environment" [35]:

- **Threats** are to be countered by the TOE, its operational environment, or a combination of the two. The definition of a threat contains:
  - Threat agents; such as hackers, users, computer processes, development personnel, administrators, and accidents. Further, a threat agent can be described by its expertise, resources, and opportunity.
  - Assets, which are violated; such as file content, content of server, or the authenticity of votes cast.
  - Averse actions, that is, influencing of one or more properties of an asset.
  - The attacker's motivation to attack the system; such as transferring unauthorised money to his account or getting knowledge about the money transfers of popular people.
  - The exploited flaw, such as the server configuration or the communication.
- **Organisational security policies (OSP)** have to be enforced by the TOE, its operational environment, or a combination of the two. Examples are security rules, procedures, or guidelines. Often legislative or regulatory bodies demand OSPs.
- **Assumptions** are defined on the operational environment to provide security functionality. Assumptions can be on physical, personnel, and connectivity aspects of the operational environment:

[4] According to the Common Criteria, OSPs are defined as "a set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment." [35] (The operational environment assists the TOE in providing its security functionality by implementing technical and procedural measures.)

- An assumption on physical aspects is for instance: “It is assumed that the TOE will be placed in a room that is designed to minimise electromagnetic emanations” [35]
- An assumption on personnel aspects is for instance: “It is assumed that users of the TOE will not write down their passwords.” [35]
- An assumption on connectivity aspects is for instance: “It is assumed that the TOE is the only non-OS application running on this workstation.” [35]

Remarks for the specification of the security problem definition:

Remark 1: If a certified system is applied in an operational environment where the assumptions do not hold, it cannot be guaranteed that the system will still provide all of its security functionality.

Remark 2: When the TOE is physically distributed, it may be better to discuss the threats, OSPs and assumptions separately for each part.

Security Objectives. The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The role of the security objectives is twofold:

- To provide a high-level, natural language solution to the previously-defined security problem.
- To divide this solution into two part-wise solutions: security objectives for the TOE and security objectives for the operational environment (these are often abbreviated with “OE.”).

Security Objective Rationale. A very important proposal of the Common Criteria is the security objective rationale where evidence to the relationship between security objectives and the security problem definition is shown. Here, it is shown which security objectives address which threats, OSPs and assumptions and that all threats, OSPs, and assumptions are effectively addressed by the security objectives, that is in detail the following:

- Each security objective traces back to at least one item from the security problem definition.
- At least one security objective is traced to each item from the security problem definition.
- Security objectives for the TOE do not trace back to assumptions (this is only possible for security objectives for the environment).
- All security objectives together achieve, that all threats are countered, all OSPs are enforced, and/or all assumptions are upheld.

Security Requirement Rationale. The security requirements rationale has a similar function than the security objective rationale. While the security objective rationale addresses the link between security problem definition and security objectives, the security requirement rationale explains why the chosen set of SFRs is appropriate and the security objectives for the TOE are correctly translated to SFRs (SFRs, which corresponds to the security objectives for the environment are not defined). If all SFRs are satisfied, all security objectives for the TOE are achieved and because of the security objective rationale also the threats, OSP and assumptions from the security problem definition are countered/enforced/upheld.

Evaluation Assurance Levels. Seven hierarchically ordered evaluation assurance levels (EAL) are defined in the Common Criteria starting with level 1 through level 7. Each EAL constitutes a subset of assurance requirements compared to the set of assurance requirements defined in the next higher level. EALs consist of an appropriate combination of assurance components from CC Part 3. All assurance dependencies of every component are addressed in the combination of levels. However, it is not required to choose one of these levels, but possible to represent other combinations of assurance components. Moreover, it is allowed to add assurance components to an existing EAL. The specific notion for such a level is called "augmentation" and corresponding levels are labelled with "+". The intention of the seven different EALs can be summarised in the following way (an overview is presented in Fig. 7.2):

- EAL1 - Functionally tested
  EAL1 is applicable where threats are not viewed as serious. Here, an evaluation only provides evidence that the TOE functions are consistent with the documentation. It is used when independent assurance is required, that is, when personal or similar information needs to be protected. An evaluation according to EAL1 includes independent testing against a specification and an examination of the user-guide. Moreover, in EAL1 it needs to be shown that the TOE is resistant to penetration attackers with a basic attack potential[5].
- EAL2 - Structurally tested
  EAL2 is applicable where a low to moderate level of independently-assured security is required, that is, to secure legacy systems. It requires the co-

EAL 1	Functionality Tested	Common Methodology for IT Security Evaluation
EAL 2	Structurally Tested	
EAL 3	Methodically Tested and Checked	
EAL 4	Methodically Developed, Tested and Reviewed	
EAL 5	Semi-formal designed and tested	Semi/ Formal Methods
EAL 6	Semi-formally verified design and tested	
EAL 7	Formally verified design and tested	

Fig. 7.2. EAL overview

[5] The attacker potential values (namely basic, enhanced-basic, moderate, and high) are further explained in the Appendix B.4.2.3 in [36].

operation of the developer (delivery of design information, test results and a vulnerability analysis). However, EAL2 does not substantially increase cost and time for the developer.

- EAL3 - Methodically tested and checked
  EAL3 is applicable where moderate level of independently assured security, a thorough investigation of the TOE and its development without substantial re-engineering is required. It requires independent and more extensive tests by the evaluator based on the functional specification and the high level design. Moreover, EAL3 demands checking the development environment.
- EAL4 - Methodically designed, tested, and reviewed
  EAL4 is applicable where a moderate to high level of independently assured security is required. Here, the developer has additional security-specific engineering costs, but EAL4 does not require substantial specialist knowledge or skills. It is the highest level at which it is likely to be economically feasible to complete to an existing product line. EAL4 is also the first level where an analysis of the source code (at least in parts) is required. Compared to EAL3, EAL4 requires more design description, the implementation representation for the entire security functions, and improved mechanisms and/or procedures providing confidence that the TOE has not been tampered with during the development process. In EAL4 it needs to be shown that the TOE is resistant to penetration attackers with an enhanced-basic attack potential.
- EAL5 - Semi-formally designed and tested
  EAL5 is applicable where a high level of independently assured security in a planned development is required. It requires a security engineering specialist. In general such a TOE is designed and developed with the intent of meeting EAL5. A modular design of the security functions is required. EAL5 is the first level where an independent vulnerability analysis is demanded. Moreover, the assumed attack potential is increased from enhanced-basic to moderate. EAL5 is also the first level where an informal language is not enough, but a semiformal design descriptions is required.
- EAL6 - Semi-formally verified design and tested
  EAL6 is applicable to TOEs for application in high risk situations where the value of the protected assets justifies the additional costs. While EAL5 demanded a semi-formal language, a formal one is required already in EAL6. In particular, a formal security model of selected security policies has to be developed. A modular and layered design of the security function is required. In addition, requirements of the development process and development environment as well as delivery procedures are defined. The assumed attack potential is increased from moderate to high.
- EAL7 - Formally verified design and tested
  EAL7 is applicable to TOEs for application in extremely high risk situations where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused se-

curity functionality. Moreover, a modular, layered, and simple design of the security functions as well as formal representation, formal correspondence, and comprehensive testing are required.

Common Methodology for IT Security Evaluation. The Common Criteria itself does not address the evaluation methodology under which the criteria should be applied. Moreover, it does not state requirements for such a regulatory framework. However, consistency between the frameworks of different evaluators is necessary to achieve comprehensive and comparable results as well as repeatability and objectivity of the results. ”An example of a regulatory framework is the CCRA (Arrangement on the Recognition of the CC Certificates in the field of IT security). This arrangement has been executed among a number of evaluation authorities in different countries and provides the conditions for mutual recognition of CC certificates between these evaluation authorities.” [35]

Another way of achieving greater comparability is using a common methodology to achieve the evaluation results. For the Common Criteria, this methodology is described in the Common Methodology for IT Security Evaluation (CEM) [36]. The CEM guides an evaluator in applying the Common Criteria. They convert the assurance requirements of the CC to concrete evaluation tasks. However, the CEM provides only comprehensive methodologies for evaluations according to the four lower evaluation levels, while only few assurance requirements added in EAL5 or even higher : EAL1 - EAL4.

Certification Process. However, some of the evaluation criteria require expert judgment and background knowledge. It is difficult to achieve this consitently. Thus, the final evaluation results are submitted to a certification process. This process serves as the independent inspection of the evaluation results. This leads to the production of the final certificate, which is usually publicly available.

The evaluation schemes and certification processes are outside the scope of the Common Criteria itself. The evaluation authorities that run such schemes and processes are responsible for these two processes. In Germany this evaluation authority is the Bundesamt für Sicherheit in der Informationstechnik (BSI - Federal Office for Information Security), which decided to use the CEM as evaluation scheme and to implement a certification processes.

Re-Evaluation. Once an evaluated and certified system is in use, it is usually enhanced, previously unknown errors or vulnerabilities may appear, or the trust model might change. Consequently, the developer might have to improve his system. Such changes may require the TOE to be re-evaluated or the security of its operational environment to be strengthened. In some instances this may only require that the needed updates are evaluated to regain confidence in the TOE. Detailed procedures for re-evaluation, including reuse of evaluation results, are outside the scope of the CC.

Protection Profile and Security Target. The CC defines two important document types: the Protection Profile (PP) and the Security Target (ST). A PP is a set of security requirements for a category of possible products/target of evaluations that meet specific consumer needs. The requirements are independent of technical solutions. The technical implementation is left open. A Protection Profile starts with a TOE overview. The main part contains the description of the TOE security problem definition, a deduced list of security objectives, corresponding security functional requirements and security assurance requirements. In addition, the security objective rationale and the security functional requirements rationale is part of the PP.

Protection Profiles go through an evaluation and certification. The evaluation is done by an accredited laboratory. An evaluation of a Protection Profile is a mainly pure document check. It simply ensures that the PP meets various syntactical and documentation rules as well as sanity checks. Therefore, the evaluator has to check whether the set of requirements is exhaustive and self-contained. Successfully evaluated Protection Profiles are certified. In Germany, this certification is done by the Bundesamt für Sicherheit in der Informationstechnik (BSI - Federal Office of Information Security). Corresponding certificates are recognised and published internationally on the Common Criteria portal.

While the PP is developed by users, a Security Target is created by the system developer, who identifies the security capabilities of his particular product. The content is equal to the one in a Protection Profile but extended by the TOE summary specification containing a description of the security functions that are implemented to ensure the security functional requirements. A ST can be more precise than a PP because it is linked to a specific TOE and not to a whole group of TOEs. A ST can be based on one or more Protection Profiles if all included PPs are certified. A Security Target evaluation is part of the whole TOE evaluation (for each EAL).

CC-Summary and Notation. Figure 7.3 shows an overview of the most of the important items of the Common Criteria. If all defined security functional and security assurance requirements are satisfied and all security objectives for the operational environment are achieved, then there exists assurance that the defined security problem is solved. That is, all threats are countered, all organisational security policies are enforced, and all assumptions are upheld. The amount of confidence in the assurance depends on the chosen evaluation assurance level.

Two aspects of the "trust model" namely the assumptions about the environment [aspect 1] and the intruder's technical capabilities [aspect 2] (see Sect. 1.2 for the whole trust model definition), can be described with the Common Criteria by defining assumptions (if wanted) and taking the intruder's capability for the threat definition into account. In addition, the intruder's capability is represented in the security assurance requirements (SARs): until EAL4 an enhanced-basic attacker potential is assumed, then for EAL5 a moderate, and for EAL6 and 7 a high attacker potential is assumed.

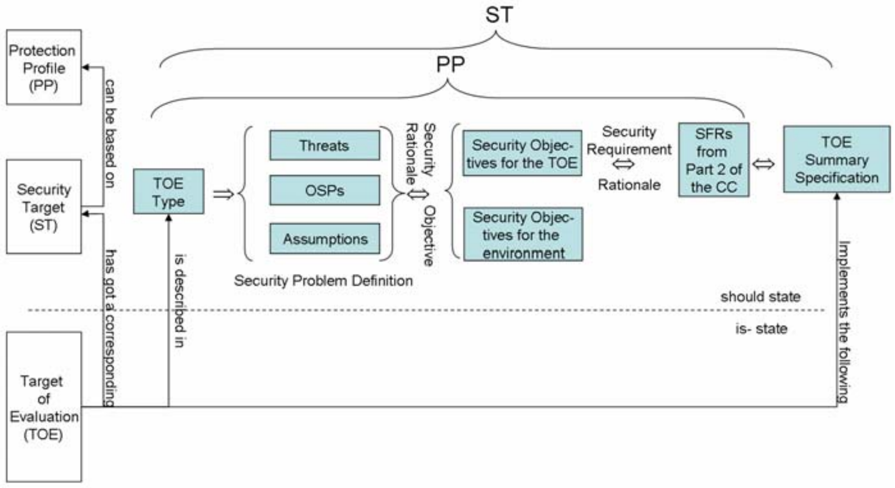

ST
PP
Protection Profile (PP)
can be based on
Security Target (ST)
has got a corresponding
Target of Evaluation (TOE)
TOE Type
is described in
Threats
OSPs
Assumptions
Security Problem Definition
Security Rationale ⇔ Objective
Security Objectives for the TOE
Security Objectives for the environment
Security Requirement ⇔ Rationale
SFRs from Part 2 of the CC
TOE Summary Specification
Implements the following
should state
is- state

Fig. 7.3. Overview of the Common Criteria

Translation: Previous Items to Common Criteria Notation. The most important Common Criteria specific items are explained above: threats, organisational security policies, assumptions, security objectives (for the TOE and for the operational environment), security objective rational, functional security requirements, and security assurance requirements. Most of these items are also used in previous chapters. However, some of these items are named different. Therefore, a translation from words used through Chap. 6 to the Common Criteria syntax is provided. Table 7.1 shows this mapping. The CC specific vocabulary is used in this book for the further parts.

The first and most important mapping is from "requirements" as used in the previous chapters to the CC "security objectives for the TOE". In particular, only the functional and security requirements comply with the security

Table 7.1. Mapping syntax from this book to CC syntax

Previous meaning of items	**Meaning according to the CC**
functional + security requirements	security objective for the TOE
threat (type, - , description, motivation)	threat (agent, asset, action, motivation)
[organisational security policy]	organisational security policy
-	assumption
assurance requirement	SAR component
1:1 mapping (treat-requirement) (OSP-requirement)	security objective rational
application note	application note

objectives for the TOE. With respect to the security problem definition, it can be said that "threats" and "organisational security policies" are also taken into account in Chap. 6, while no "assumptions[6]" have been defined. Assumptions are consciously left out in Chap. 6, in order to provide the maximised security because, here, the environment does not need to be trustworthy in the sense of implementing any kind of security functionality.

In previous chapters, threats are named explicitly and mapped to a corresponding security requirement (security objective for the TOE), while organisational security policies are only implicitly named because they would be a simple reformulation of corresponding functional requirements (security objectives for the TOE). This corresponds to the CC security objective rational, even though, the Common Criteria does not demands a 1:1 mapping.

The Common Criteria defines for each threat: the "threat agent", "asset", "adverse action", and the "motivation". These items can be deduced from each threat in Chap. 6, where the intruder's type and motivation are named and his action described. The adverse assets can in general be deduced from the motivation.

The item "assurance requirements" corresponds to "security assurance requirements". The difference is that the one defined in Sect. 6.5 is based on the standardised language defined in Chap. 4 while the CC provides its own language in terms of predefined components. Application notes are used in both the CC and the previous chapters with the same meaning.

7.2 Discussion of Possible Trust Models

Chapter 6 provides a long list of security and functional requirements for remote electronic voting systems. If one develops such a system, it would not be based on any security functions of the environment the system is used in. The system could be used in a mainly insecure environment, and nevertheless, such a remote electronic voting system could be used to run a secure election. However, developing a system in compliance with these requirements is a rather difficult, if not unfeasible, task. Moreover, such a system would become very complex because it needs to contain many more components than the actual voting software to run the voting protocol. It is likely that it would be too complex to be evaluated (in reasonable time and under justifiable costs). Therefore, the idea is (not only for electronic voting systems, but in general) to not use the remote electronic voting systems in a completely insecure environment, but where some assumptions about the environment hold and the intruder's technical capabilities are restricted. This concept is supported by the Common Criteria's security problem definition in combination with the

[6] A few assumptions about personal aspects in terms of the Common Criteria are already identified in Sect. 6.2, namely A.ProtectedEnvironment, A.ProperConfig, and A.AuthToken.

definition of the intruder's technical capability by choosing the correspondent AVA_ VAN component.

Depending on the environment and the importance of the election (and, thus, the expected intruders), abstractly spoken, some of the threats listed in Chap. 6 can be completely, or in parts, removed by corresponding assumptions about the environment. According to the Common Criteria, those security objectives for the TOE can be shifted to those for the environment. These identified assumptions about the environment compose the definition of the first aspect of the trust model (see Sect. 1.2 for the whole trust model definition). Here, the responsible election authority needs to decide whether these assumptions fit their environment and whether they can ensure the security objectives for the environment. Therefore, the list of assumptions can vary from election to election. However, there is a maximised set of assumptions which should not be extended for any type of elections. This set is part of a maximised trust model that is discussed in Sect. 8.2.3.

Obviously, the definitions of security objectives for the TOE versus those for the environment have consequences for the system design and complexity. In the following two subsections, two different trust model aspects and their consequences for the remote electronic voting system are discussed[7]:

- The first example concerns O.T.TamperClient as a security objective for the environment for the TOE.
- The second example concerning the intruder's capability of an attacker sniffing on the network (O.T.ElecSecrecyNet) and whether he is able to decrypt encrypted votes or not.

7.2.1 Trustworthy Vote Casting Device

With respect to the trustworthiness of the vote-casting device, Chap. 6 contains the following security objective for the TOE with its corresponding threat:

O.T.TamperClient: The ·*client-side voting software*· `shall` ensure that its operations and data are unaffected by other applications running on the ·*vote-casting device*·.

T.TamperClient: An outside intruder runs malware on the ·*vote-casting device*·, which either reads the ·*vote*· (in order to compromise the secrecy of the vote), alters the ·*vote*·, or reads the ·*authentication information*· to ·*cast*· a ·*vote*· or to bar the ·*voter*· from ·*casting a vote*· (in order to affect the election result).

Defining the trust model, the responsible election authority needs to define with respect to O.T.TamperClient, whether the corresponding threat is already prevented by the environment, that is, it does not exist in their sce-

[7] This part has partly already been published in [146].

nario at all[8], or has to be prevented by the remote electronic voting system. Depending on the decision, the following three cases can be distinguished:

Case 1: The responsible election authority can assume that voters use a trustworthy vote-casting device. This could be the case, for instance, for staff and council work elections where all PCs are centrally administered and secured. Thus, intruders have no possibility of installing malware on purpose and the administrator[9] can implement adequate security mechanisms on the PCs. Thereby, the administrator can ensure that all running applications do not interfere with the client-side voting software. For this case, the named threat can be reformulated to a corresponding assumption (on connectivity aspects) to the environment:

A.TamperClient: The ·*vote-casting device*· is trustworthy.

According to the Common Criteria practise, the security objective is adjusted: it does not need to be ensured by the client-side voting software but by the environment:

OE.TamperClient: The administrator(s) of the ·*vote-casting device*· is responsible for its trustworthiness, that is, ensuring that other applications running on the ·*vote-casting device*· do not interfere with the ·*client-side voting software*·, its operations, and its data as well as preventing any intruder from running malware on the ·*vote-casting device*· which interferes with the ·*client-side voting software*·, its operation, and its data.

Consequences: Now, the developer of a remote electronic voting system can assume a trustworthy vote-casting device and does not need to implement any security functions to protect the client-side voting software. However, such a remote electronic voting system only enables secure elections if the assumption holds for the client-side voting software, that is, if the administrator successfully secures the vote-casting device.

Case 2: The responsible election authority assumes for their election no such attack as described in T.TamperClient to their election. A possible reason for such a decision can be that the effort to implement such specific malware is too much effort compared with the value of the election. Thus, the probability for the appearance of the threat is negligible. However, it is well-known that various kinds of malware are available and many vote-casting devices are already infected. Such malware usually tries to interfere with e-banking applications or, in general, to catch user logins and corresponding passwords.

[8] This can also be the case if the probability for the appearance of the threat is negligible.

[9] Note, depending on the scenario, this administrator can also be a subgroup of the inside intruder. In this case, the situation would be vise versa and case 3 eventuates.

To handle this remaining part of T.TamperClient, either a corresponding assumption of the environment needs to be defined (case 2A) or this part remains (case 2B):

Case 2A: The following two assumptions can be distinguish:

A.TamperClientA: Any intruder does not try to run remote electronic voting specific malware on the ·*vote-casting device*· which interferes with the ·*client-side voting software*·, its operation, or its data.

A.TamperClientB: The ·*vote-casting device*· is trustworthy with respect to standard vulnerabilities..

The corresponding security objectives for the environment are:

OE.TamperClientA Remote electronic voting specific malware which interferes with the ·*client-side voting software*·, its operation, or its data does not exist on the ·*vote-casting device*·.

OE.TamperClientB The ·*voter*· is responsible for the trustworthiness of his ·*vote-casting device*· with respect to standard vulnerabilities.

Consequences: The developer of a remote electronic voting system can assume a trustworthy vote-casting device and does not need to implement any security functions to protect the client-side voting software. However, such a remote electronic voting system only enables secure elections if the assumption holds for the client-side voting software, that is, remote electronic voting specific malware does not exist and the voter secures his vote-casting device against standard vulnerabilities. To do so, he must have the ability to secure his vote-casting device. Thus, the responsible election authority shall help voter clean his vote-casting device from such malware. The Gesellschaft für Informatik (GI - the German society of computer scientists), for instance, applies for their elections a simplified voters' guide [51], which contains one page of general hints and thirteen easy-to-follow one-sentence rules for voters. A similar approach has been taken by the Swiss electronic voting project.

Case 2B: For this case, the security problem definition contains, compared to case 2A, only the first assumption, while the second assumption is replaced by a threat:

A.TamperClient: Any intruder does not try to run remote electronic voting specific malware on the ·*vote-casting device*· which interferes with the ·*client-side voting software*·, its operation, or its data.

T.TamperClient: An outside intruder uses standard malware on the ·*vote-casting device*·, which either reads the ·*vote*· (in order to compromise the secrecy of the vote), alters the ·*vote*·, or reads the ·*authentication information*· to ·*cast*· a ·*vote*· or to bar the ·*voter*· from ·*casting a vote*· (in order to affect the election result).

From this assumption and threat, the following security objective for the environment and for the TOE can be deduced:

OE.TamperClient Remote electronic voting specific malware which interferes with the ·*client-side voting software*·, its operation, or its data does not exist on the ·*vote-casting device*·.

O.T.TamperClient: The ·*client-side voting software*· `shall` be robust against standard vulnerabilities[10] of ·*vote-casting device*·.

Consequences: The developer of a remote electronic voting system cannot assume anymore that their client-side voting software runs on a completely trustworthy vote-casting device but where standard vulnerabilities still exists. Thus, the developer needs to be aware of standard vulnerabilities and must demonstrate that he implements the corresponding security functionality.

Case 3: The responsible election authority considers the vote-casting device as open systems and assumes that voters are not able to protect themselves efficiently against malware. Moreover, from their opinion, it cannot be excluded that a malicious voter manipulates his vote-casting device on purpose, in order to generate a proof of his choice, since a platform owner has complete control over it. Thus, the security problem definition remains as defined in chapter 6.

Consequences: In this case, T.TamperClient produces a serious problem because malicious code can be distributed easily and automatically, for example, by exploiting security flaws of the vote-casting device or by sending infected e-mails to voters, which could be done massively via viruses. Malicious code could also be put on the vote-casting device by developers of products running on many vote-casting devices (for example, Solitair). Compared to postal voting, this attack can be done automatically and in large-scale with significant impact on the election result.
Moreover, common cryptographic means do not overcome any of these two attacks, since malicious code can interact before the cryptographic operations are applied. The intruder may, for instance, eavesdrop on mouse or keyboard inputs and deduce the voter's choice.

[10] In order to use this case, in a Protection Profile, standard vulnerabilities needs to be further specified, for instance by using the Common Criteria security functional requirement componente FPT_ TEE .

Different approaches to implement a security function to meet the case 3 security problem definition have been proposed in the past, while most of them address the problem but do not satisfactorily solve it for the described security problem definition (case 3)[11]:

- The Swiss and GI guidelines explaining to voters how to improve the trustworthiness of their vote-casting device: this approach can reduce the risks created by malware, but many voters are not likely to be able to follow the instructions. Moreover, such an approach is useless against malicious voters installing malware on purpose.
- Otten proposes in [115] a special voting operating system based on Knoppix. Here, voters have to boot their vote-casting device from CD. This approach also does not solve the malicious voter problem, but it prevents attacks caused by malware.
- Schweisgut proposes in [135] and Juels et al. in [77] the application of an observer, for instance, a smart card. By doing so, they overcome most of the attacks from malicious voters. However, a smart card does not interact directly with the voting server but over the vote-casting device. Malware on this device can mount a man-in-the-middle attack and misuse the card, for instance, by sending a wrong candidate choice to the smart card or the vote-casting device displays a modified ballot.
- Helbach et. al propose in [64] (and the later improved version [114]) the code sheets to overcome the problem with malicious clients. This code sheet is sent via ordinary mail and contains for each candidate a voting TAN and a confirmation TAN[12]. The voter enters a corresponding voting TAN instead of choosing a candidate on the PC screen. To verify the correctness, he compares the received and displayed confirmation TAN with the one on the code sheet. The disadvantages of this approach concerns the user-friendliness (which decreases in particular for complex ballots implementing) and the fact that the requirement O.T.ProofGen can only be ensured if vote updating is applied.
- Another approach proposes to use an appropriate security architecture based on a security kernel and on Trusted Computing elements. Such a solution is the only one that could efficiently prevent the described threat. However, currently, there are still open problems with Trusted Computing and it is not easy to know-how to integrate the Trusted Computing elements in a Common Criteria evaluation. For a more detailed discussion of this case and in particular the Trusted Computing based approach, see [144] and [4].

This short analysis shows that currently defining only a security objective for the TOE with respect to the client weakness would avoid the appli-

[11] However, the provided solutions can be used by the voter or the administrators in the other proposed cases.

[12] To overcome vote selling the authors introduced in [114] an additional TAN – the so called finalisation TAN.

cation of remote electronic voting systems because the only approach meeting the security objective is not yet implemented and ready for a large-scale application, such as in an election.

7.2.2 Compromising Encryptions

There exists another security objective in Chap. 6, which is difficult to meet depending on the concrete definition of the trust model in terms of the intruder's technical capabilities:

O.T.ElecSecrecyNet: The ·*remote electronic voting system*· `shall` not provide any information in transmitted protocol messages, which allow one to construct the link between a particular ·*voter*· and his ·*vote*·. The ·*remote electronic voting system*· shall ensure that neither the ·*vote*· itself nor the number of chosen ·*voting options*· (including an empty ·*ballot*·), nor a ·*spoilt*· ·*vote*· (for example, by using the length of the protocol messages) can be linked to a particular ·*voter*·. In addition, it `shall` be ensured that the sequence of messages does not reveal the link.

T.ElecSecrecyNet: An outside intruder sniffs the network in order to compromise the secrecy of the vote.

With respect to this security objective, the responsible election authority needs to decide ...

Case 1: ... whether it is acceptable that the intruder is able to compromise the secrecy of the vote after a particular point in time (for instance, after the next election). The consequence is that the named threat needs to be extended in the following way:

> **T.ElecSecrecyNet:** An outside intruder sniffs the network in order to compromise the secrecy of the vote before the next ·*election*·.
>
> A similar extension needs to be added to the security objective:
>
> **O.T.ElecSecrecyNet:** The ·*remote electronic voting system*· `shall` not provide any information in transmitted protocol messages, which allow one to construct the link between a particular ·*voter*· and his ·*vote*· before the next ·*election*· [...].

Case 2: ... whether it is acceptable that the intruder is able to compromise the secrecy of the vote (either before or after the next election) as long as he cannot prove the link to a third party (for instance, because he knows: $vote_i$ and $sig_{voter}(enc(vote_j))$ and can prove that $vote_i = vote_j$ while he can also show that $sig_{voter}(enc(vote_j))$ was cast, for instance, because it is stored on a bulletin board). The consequence is that the named threat needs to be changed in the following way:

> **T.ElecSecrecyNet:** An outside intruder sniffs the network in order to compromise the secrecy of the vote and is able to prove the link between the voter and his vote.

A similar modification needs to be added to the security objective:

O.T.ElecSecrecyNet: The ·*remote electronic voting system*· `shall` not provide any information in transmitted protocol messages, which allow one to construct the proof for the link between a particular ·*voter*· and his ·*vote*·. The ·*remote electronic voting system*· shall ensure that neither the ·*vote*· itself nor the number of chosen ·*voting options*· (including an empty ·*ballot*·), nor a ·*spoilt*· ·*vote*· (for example, by using the length of the protocol messages) can be used to generate a proof for the link between a particular ·*voter*·. In addition, the sequence of sent and received messages does not reveal a proof.

Case 3: ... whether only those remote electronic voting systems are acceptable, which ensure that the voter can never be linked to his vote by an outside intruder sniffing the network (this would be in compliance with the temporal unlimited secrecy of the vote demanded in [141]). Thus, the security problem definition remains as defined in Sect. 6.3.

Case 4: ... whether it is acceptable that an intruder can compromise the secrecy of a vote for an arbitrary voter as long as he cannot compromise it for a particular voter. That is, the intruder cannot decide before the election that he wants to know-how person X casts his vote. This case is not further discussed in this book because there is no motivation for the responsible election authority to choose this case.

Feasibility Study for Case 1 - Case 3. In general, it is not possible to prevent network sniffing even sniffing and data decryption are punishable with § 202 a StGB in Germany and corresponding laws in other countries. The intruder works in the following way: he sniffs all voting protocol messages transmitted to the voting server, stores these data in a database and analyses them later. These messages are encrypted with state-of-the-art encryption algorithms which are classified as secure. The problem with respect to the security objective O.T.ElecSecrecyNet is that the chosen algorithms might be classified as secure for the present and maybe also the near future but no statements for the long future can be made. Perhaps, someone will find a fast algorithm to decrypt messages without the knowledge of the secret key, allowing to compromise the applied cryptographic algorithm. In any case, by using adequate computational power, single messages can be decrypted or single secret keys can be calculated (brute force trials). Depending on the intruder's computational power, the intruder will be in a position to decrypt all or at least single encrypted ballot messages somehow in the future. Thus, it cannot be prevented that, in some way, the intruder is able to decrypt these messages in the long future. However, to ensure O.T.ElecSEcrecyNet [case 1] the application of state-of-the-art encryption algorithms would work.

The question to be answered for the other two cases is, whether the intruder is able to link the decrypted vote message to the corresponding voter

ID or whether he only gets decrypted vote messages but cannot link these to voters.

The analysis of different types of remote electronic voting systems in [152] and [157] shows that temporal unlimited secrecy of the vote as demanded in O.T.ElecSecrecy-Net [case 3] cannot be ensured by any of the analysed remote electronic voting systems. This is caused by the available link between voter and his IP-address, or even more, because of the application of a voter digital signature on the (encrypted) vote like in Estonia [94]. Exceptions are those remote electronic voting systems that implement vote updating like in Estonia or those that implement a two-phase voting protocol[13] (see, for instance, [82]). Moreover, [152] points out that, in general, there is no possibility to prove the knowledge to a third party, which enables the application of O.T.ElecSecrecyNet [case 2].

7.3 Evaluation Assurance Level According to the Requirements

The assurance requirements from Sect. 6.5 have been developed based on the authors own experiences and assurance requirements from existing literature. This section translates the assurance requirements from Sect. 6.5 to the security assurance requirements (SARs) components from the CC framework. This is not easy and does not always have an unambiguous mapping. The SARs are structured in the following way: for each security assurance requirement component, there is a set of requirements for 'developer action elements', 'content and presentation elements' and 'evaluator action elements'. In contrast, each of the security assurance requirements defined in previous chapters addresses only one of these three groups. Thus, in general, more than one security assurance requirement from Sect. 6.5 can be mapped to one SAR component. Moreover, the previously defined security assurance requirements are very abstract; they depend on the specific interpretation of which component of a particular SAR family is chosen for the mapping (the components are distinguished by their strength). For instance, for some requirements, it must be decided whether the demanded property must only hold for parts of the system or for the whole system. For the mapping in this book, those components that demand completeness have been chosen. Table[14] 7.2 illustrates the mapping.

Three security assurance requirements could not be mapped according to Common Criteria components: Assur.2 (demanding reliable components),

[13] The implementation of a two-phase protocol would be similar to postal voting. The voter has a special period of time to ask for her anonymous token and in a second period of time she can cast the ballot.

[14] Assur. 3a is missing in the table. However, it addresses the documentation from Assur. 4 which is covered. Thus, Assur. 3a is covered, too.

Assur.7 (which is a non-core component) and Assur.12 (discussing the trust model compliance). All three are outside the scope of the Common Criteria.

This list of SAR components is preferable to the very high level security assurance requirements from Sect. 6.5. The SARs from Table 7.2 are much more detailed and complete. During the mapping, it turned out that the security assurance requirements from Sect. 6.5 are not complete. For instance, the source code and a system specification from the electronic voting system are necessary, but the mapping from the system specification to the source code, which is really essential, is missing. By introducing dependencies for SAR components, this vulnerability is avoided.

Mapping the identified SAR components to the seven evaluation assurance levels (EALs), the list mainly corresponds to **EAL4 +**[15]. However, those SAR components related to the security target are missing (namely, ASE_ CCL.1, ASE_ ECD.1, ASE_ INT.1, ASE_ OBJ.2, ASE_ REQ.2, ASE_ SPD.1), and correspondingly, the dependencies of ASE_ TSS.1 are not satisfied. This is not surprising as during the development of the security assurance requirements from Sect. 6.5, the Common Criteria and, thus, the concept of a security target was not taken into account. According to the SARs listed in Table 7.2, EAL4 is augmented in the following way:

- ADV_ IMP.2 instead of ADV_ IMP.1
- add ADV_ INT.3
- ALC_ CMS.5 instead of ALC_ CMS.4
- add: ALC_ FLR.1
- ATE_ COV.3 instead of ATE_ COV.2
- ATE_ IND.3 instead of ATE_ IND.2
- AVA_ VAN.4 instead of AVA_ VAN.3

There are obviously a couple of augmentations. However, to reach EAL5, the following SAR components are missing (mainly those addressing the application of semi-formal methods):

- ADV_ FSP.5 (Complete semi-formal functional specification with additional error information) instead of ADV_ FSP.4 (Complete functional specification)
- ADV_ TDS.4 (Semiformal modular design) instead of ADV_TDS.3 (Basic modular design)
- ATE_ DPT.3 (Testing: modular design) instead of ATE_ DPT.2 (Testing: security enforcing modules)

However, the augmentation also includes components which would also augment EAL5: ADV_ IMP.2, ADV_ INT.3, ALC_ FLR.1, ATE_ COV.3, ATE_ FUN.2, and ATE_ IND.3. This clarifies that, in deed, the recommended level is much more than EAL4 while the reason for not calling it EAL5+ is the missing demand for semi-formal methods in the set of requirements from Sect. 6.5.

[15] This corresponds to Mercuri's opinion that EAL4 is the lowest level that should be applied to certify electronic voting systems [101].

Table 7.2. Mapping: SARs - security assurance requirements (from Sect. 6.5)

SARs	**assurance requirements (sect. 6.5)**
ADV_ ARC.1 - Security architecture description	Assur.4 b)
ADV_FSP.4 - Complete functional specification	Assur.4 a)
ADV_ IMP.2 - Implementation of the TSF	Assur.3 c), Assur.10 c)
ADV_ INT.3 - Minimally complex internals	Assur.5
ADV_ TDS.3 - Basic modular design	Assur.4 d)
AGD_ OPE.1 - Operational user guidance	Assur.4 h)
AGD_ PRE.1 - Preparative procedures	Assur.4 e) i), AS.11 a)
ALC_ CMC.4 - Production support, acceptance procedures and automation	Assur.1 b), Assur.4 e), Assur.10 e)
ALC_ CMS.5 - Development tools CM coverage	Assur.1 b), Assur.4 e), Assur.10 e)
ALC_DEL.1 - Delivery procedures	Assur.4 i), Assur.11 a)
ALC_DVS.1 - Identification of security measures	Assur.4 g), Assur.11 b)
ALC_ FLR.1 - Basic flaw remediation	Assur.1 c), Assur.3 d), Assur.10 d)
ALC_ LCD.1 - Developer defined life-cycle model	Assur.1 a), Assur.10, Assur.11 c)
ALC_ TAT.2 - Compliance with implementation standards	Assur.1 a), Assur.11 c)
ASE_ TSS.1 - TOE summary specification	Assur.4 c)
ATE_ COV.3 - Rigorous analysis of coverage	Assur.4 f), Assur.6
ATE_ DPT.2 - Testing: security enforcing modules	Assur.4 f), Assur.6
ATE_ FUN.1 - Functional testing	Assur.4 f), Assur.6
ATE_ IND.3 - Independent testing - complete	Assur.3 b), Assur.10 b), Assur.9
AVA_ VAN.4 - Methodical vulnerability analysis	Assur.8

Surprisingly, none of the analysed catalogues takes semi-formal or even formal methods for the evaluation of electronic voting systems into account.

7.4 Formal IT Security Model

The analysis result from the previous section (EAL4 +) corresponds to the system evaluations based on the Common Criteria from the past: most evaluations have been done based on evaluation assurance levels equal or below EAL4+, since starting from the EAL5 semi-formal and/or formal methods are required. The application of such methods causes substantial additional effort for manufacturers and evaluators. The decision for such a high evaluation assurance level should be made before starting the development because (semi-) formal methods cannot be implemented in the follow-up (the effort to do so in the follow-up is as large as a complete new development). However, political elections are the highest property of a democracy and if formal methods are not applied for these applications where then (compare to [147] and [148])?

Moreover, EAL5 provides a substantial increase in the trustworthiness of certified systems compared to EAL4, because a semi-formal description of the system design as well as a more modular and therefore better analysable architecture is demanded. A corresponding increase can be identified from EAL5 to EAL6 because the semi-formal specification languages are replaced by formal specification languages. “Past experiences show that a formal modelling of the security policies given as a formal security model may lead to an increase of confidence in the security of the product that obeys these security policies.” [97].

Advantages. The application of formal IT security models has three main advantages:

- No natural language can guarantee an unambiguous interpretation and, therefore, it provides no feasibility to prove consistence in the formulation of secure states and permitted state transitions. Vulnerabilities in the implementation of these are a consequence. In contrast, the application of mathematical established technical equipment, which makes the application of computer-aided proofs possible, enables the definition of unambiguous and intersubjective secure states and permitted state transitions.
- The development of a formal IT security model is used to identify and remove inconclusive, inconsistent, contradictory, or not enforceable secure states and/or permitted state transitions which cannot be detected with natural language.
- Using natural language for the specification of secure states and permitted state transitions causes similar problems for the evaluator: it is hard and in general not unambiguous to decide whether the implemented security functions are sufficient to ensure the specified secure states and permitted state transitions. Based on a formal specification of the system, it can

be formally proven that the specification and later the implementation conform to the formal specification of the secure states and permitted state transitions.

Starting from EAL6, the Common Criteria component ADV_ SPM.1 has to be ensured. It demands the use of a formal IT security model. Moreover, this component requires a consistency proof (in form of a mathematical proof) for the model itself and a compliance conformance between the system specification and the defined model. To do so, it is possible to use already published and established formal IT security models[16] - as a whole or in parts. If no suitable formal IT security model exists, such a model must be developed.

The latter case holds for remote electronic voting systems. Therefore, such a formal IT security model has to be developed before an evaluation according to EAL6 and/or 7 can be aimed. In this section, it is shown, by the example of some concrete security objectives defined in the Protection Profile, how such a formal IT security model can be designed.

In the further parts of this section, the definition of an IT security model is introduced (see Sect. 7.4.1), then it is discussed whether existing IT security models can be applied (see Sect. 7.4.2). Subsequently, security objectives from Sect. 6.3 are identified, which are considered for the definition of a formal IT security model (see Sect. 7.4.3), and afterwards a formal IT security model is developed and proven to ensure all characteristics of an IT security model (see Sect. 7.4.4)[17].

7.4.1 General Introduction

Model Definition. According to [58], IT security models define system states and state transitions, differentiate between secure and insecure states, and explain under which circumstances secure states are reached. An IT security model can be more or less formal. All IT security models contain the following *five description elements*:

1. The definition of a superior security objective
2. The specification of secure system states [18] which represent together the superior security objective
3. A trust model, describing a set of assumptions about the environment in which the system is used and under which the set of secure system states is equivalent to the superior security objective
4. A set of permitted state transitions
5. A security theorem, claiming that applying any permitted state transitions to any secure state necessarily transfers to a secure state again

[16] Examples for available and established IT security models are: Bell/LaPadula model [10], the Clark Wilson model [33], and the Biba model [13].

[17] These parts have been published in [56] and [57].

[18] The specification of secure system states corresponds to the Common Criteria security objectives (in the case of a non formal IT security model).

Explaining the Coherences. An IT security model has to close the following two gaps:

- The gap between the secure system states and the superior security objective (trust model in 3)
- The gap between the permitted state transitions and the secure system states (security theorem in 5)

The first gap can be closed by a Protection Profile; in particular by

- the security problem definition, including a list of assumptions about the environment,
- the list of security objectives for the system, and
- the discussions in section "security objective rationale".

As the specification of a Protection Profile is discussed in Chap. 8, this aspect is not further discussed in this section. The second gap is closed by the security theorem with its corresponding proof.

Definition of Secure System States and Permitted State Transitions. The secure states (description element 2) and the permitted state transitions (description element 4) have to be described as accurate and precise as possible. One informal way to formulate secure states is the definition of security objectives according to the Common Criteria [35]. In this case, the security theorem (description element 5) is proven by a linguistically convincing and conclusive argumentation. For applications which require a high security assurance the definitions of a secure state and of permitted state transitions must be consistent and the corresponding security theorem must hold without any doubt. In this case, it is necessary to specify the secure states and the permitted state transitions in a formal way, and the security theorem must be proven with mathematical means. The formal specification of both together (in description elements 2 and 4) together with the formal proof (in description element 5) represents a formal IT security model[19].

Note, in the case of a formal IT security model a third gap has to be closed - the gap between the linguistically formulated security objectives from the Protection Profile and the formal specification of the secure states. This cannot be formalised, but this is the subject of an argumentative discourse of security and application experts.

[19] The Common Criteria defines formal security models in the following way: "A formal security model is a precise formal presentation of the important aspects of security and their relationship to the behaviour of the TOE; it identifies the set of rules and practises that regulates how the TSF manages, protects, and otherwise controls the system resources. [...] the formal security policy model is merely a formal representation of the set of SFRs being claimed." [35]

7.4.2 Application of Available IT Security Models for Elections

To the author's knowledge, no formal IT security model is available which completely covers the superior security objective of a secure remote electronic election. Caused by the numerous different tasks of a remote electronic voting system, the existence of such a model also seems to be unrealistic. However, the integrity model of Clark Wilson [33] and the confidentiality model of Bell-LaPadula [10] can possibly describe partial security objectives.

The Clark Wilson model introduces the separation of duty principle to security modelling. For different partial security objectives in the context of a remote electronic voting system, it might be possible to use the separation of responsibilities in the sense of Clark Wilson. Section 6.4.1 demands, for example:

O.T.SepDuty: The access control mechanism `shall` only allow access to the *·voting server·* if at least two different *·users·* are logged in.

This security objective (requirement) corresponds to the certification rule C3 and the penetration rules E2 and E3, which describe the "internal consistency" of a system in the Clark Wilson model:

- E2: The system has a list mapping users to transaction procedures (user X, TPi, (CDIa, CDIb, CDIc, ...)) and ensures that users can only execute transaction procedures according to this list.
- C3: The allocation list from rule E2 complies with the separation of duty principle.
- E3: The system authenticates the user's identity before executing any transaction procedure.

The Bell-LaPadula model prevents confidential information flow to public domains. This is achieved by mandatory access control. This approach could conceivably structure voters, poll workers, ballots, and the e-ballot box in a hierarchical information flow model a là Bell-LaPadula and, thus, to model the secrecy of the election. These approaches are still open research tasks.

The following subsections discuss other security objectives (requirements) from Sect. 6.3 and 6.4, which cannot be modelled with Bell LaPadula, Clark Wilson or one of the other well-known formal IT security models. Therefore, a new formal IT security model is developed for these security objectives. The developed transaction procedures for the penetration of these security objectives could be embedded into a superior separation of duty model according to Clark Wilson. This integration needs to be further analysed in the context of future work.

7.4.3 Selection of Security Objectives

The development of a formal IT security model for remote electronic voting systems is a complex task and happens gradually by adding security objectives step by step. The security model, which will be presented in Sect. 7.4.4, is a first step accomplished for three selected security objectives (requirements) from Sect. 6.3 and 6.4. This first step illustrates how the further security objectives can be specified formally. The three selected security objectives are:

O.T.IneligVotes: The *·remote electronic voting system·* `shall` store in the *·e-ballot box·* only *·e-votes·* cast from *·eligible voters·*. [...]

O.T.OneVoterOneVote: The *·remote electronic voting system·* `shall` store in the *·e-ballot box·* only one *·vote·* per *·voter·*. In particular it `shall` store the first received *·vote·* per *·voter·*.

O.OSP.VoteRight:The *·remote electronic voting system·* `shall` ensure that no *·voter·* looses his voting right without having *·cast a vote·*.

7.4.4 Formal IT Security Model for Remote Electronic Voting

Different possibilities to model a particular system exist. According to [58] an IT security model for the above identified security objectives can be described in the following way:

(1.) Definition of the Superior Security Objective

Execution of a secure, equal, universal, direct, secret, and free remote electronic election.

Definition of a System State. A state is represented by a triple of the following three entries:

- W - Set of eligible voters (those who are listed in the electoral register and have not yet cast a vote).
- S - Set of (encrypted) votes stored in the e-ballot box.
- $voter : S \rightarrow M$ - Mapping (encrypted) votes on their electors.
  M is a superset of W_{total}, that is, $M \supseteq W_{total}$. M contains any user who tries to access the remote electronic voting system(independent whether or not this particular user has the right to cast a vote). The function *voter* assigns each (encrypted) vote to its producer (elector).

  Remark 1: In the case of postal voting, this function *voter* is realised by the outer envelope which is labelled with the sender's name and address. During the tallying phase, the sender information is checked and it is verified whether the sender has the right to cast a vote (this is the moment when a link between the voter and his "encrypted" vote exists). If he has the right to cast a vote, the outer envelope is removed, the inner one containing the vote is put into the ballot box, and the corresponding voter is flagged in the electoral register. In the ballot box, the votes are anonymously stored but you know that $voter(s) \in W_{total}$ holds because of the verification with the outer envelope.

> *Remark 2:* The values of *voter* are visible only for the last vote (or votes) cast into the e-ballot box, that is, only for the $s \in S_{i+1} \backslash S_i$. After anonymising S, the values of *voter* cannot be reconstructed. Therefore, in praxis, the *voter* mapping should only be used during state transitions on the $s \in S_{i+1} \setminus S_i$. Secure state transitions are controllable on this "visible subset" $S_{i+1} \setminus S_i$ of S_{i+1} only (see rules for permitted state transitions (4) below). For the "invisible part" S_i of the *voter* mapping on S_{i+1}, the following is defined: $voter_{i+1}|S_i := voter_i$.

Initial State. $\langle W_0 = W_{total}, S_0 = \{\}, voter_0 = \{\}\rangle$ is the initial state. W_{total} stands for the set of all voters in the electoral register (those who have already cast a vote and those who still have the right to cast a vote). The two empty sets S_0 and $voter_0$ stand for the empty e-ballot-box in the beginning and the corresponding empty mapping of the empty box on the users of the voting system.

(2.) Specification of Secure States

It has to be defined which properties represent a secure state. According to Sect. 7.4.3, the security objectives O.T.IneligVotes, O.OSP.Vote-Right, and O.T.OneVoterOneVote are selected to be specified in terms of formal state properties denoting a secure state.

- O.T.IneligVotes: $\forall s \in S : voter(s) \in W_{total}$, that is, the e-ballot box contains only those e-votes ($s \in S$) from which the corresponding elector ($voter(s) \in W_{total}$) is listed in the electoral register. In order to ensure this, the voter needs to be unambiguously identified and authenticated.
- O.T.OneVoterOneVote: $\forall s, s' \in S : voter(s) = voter(s') \Rightarrow s = s'$, that is, whenever the set S of cast votes contains two votes from the same voter then these two votes are identical. Thus, only one of the stored e-votes is tallied. This means that each voter can cast only one vote.
- O.OSP.VoteRight: $\forall x \in W_{total} \setminus W : \exists s \in S : voter(s) = x$, that is, a voter can only become an elector if his e-vote is stored in the e-ballot box ($s \in S$). Thus, he cannot lose his right to vote without having cast a vote which has been successfully stored in the e-ballot box.

Remark: It is easy to prove that these three conditions for a secure state are equivalent to the following two conditions: "$W_{total} = W + voter(S)$" (where "+" denotes the disjoint union of sets) and "The *voter* mapping is injective". An alternative way to prove the security theorem (5) would be to prove that these two conditions are implied by the permitted state transitions (4). However, we prefer to derive our three conditions of a secure state (2) directly from the following permitted state transitions.

(3.) Trust Model

The set of assumptions about the environment and the corresponding reasoning are part of [161].

(4.) Permitted State Transitions

A state transition from state $Z_i = \langle W_i, S_i, voter_i \rangle$ to $Z_{i+1} = \langle W_{i+1}, S_{i+1}, voter_{i+1} \rangle$ is permitted if one of the following two rules[20] holds:

- State transitions in which no vote is cast:
  [rule 1] $W_i = W_{i+1} \wedge S_i = S_{i+1} \wedge voter_i = voter_{i+1}$
- State transitions in which a vote is cast and successfully stored in the e-ballot box, that is, the sets S and W are modified:
  [rule 2] $\exists s \in S_{i+1} : (voter_{i+1}(s) \in W_i \wedge W_{i+1} = W_i \setminus \{voter_{i+1}(s)\} \wedge S_i = S_{i+1} \setminus \{s\})$

Remark 1: All $m \in M$ can initiate a state transition by casting a vote. However, for not permitted state transitions holds: $m \in M \setminus W_{total} \Rightarrow W_{i+1} = W_i$ and $S_{i+1} = S_i$.

Remark 2: The state transition rules use the *voter* mapping only on its visible part, that is, on $S_{i+1} \setminus S_i$. This makes the transition rules usable in praxis.

(5.) Security Theorem

For all permitted state transitions starting with the initial state $Z_0 = \langle W_{total}, \{\}, \{\} \rangle$ holds that any reachable state is a secure state.

Proof. The security theorem can be proven by mathematical induction. To simplify the notation, *voter* is used instead of $voter_{i+1}$ or $voter_i$, while $voter_{i+1}|S_i := voter_i$ holds. To simplify the main proof it is helpful to first prove that for all permitted state transmissions Z_0 to Z_i the following three lemmas L1, L2, and L3 hold. These are now named and proven:

L1: $S_i \neq S_{i+1} \vee W_i \neq W_{i+1} \Rightarrow \exists s \in S_{i+1} : (S_{i+1} \setminus S_i = \{s\} \wedge W_i \setminus W_{i+1} = \{voter(s)\})$

Interpretation: During each permitted state transitions according to [rule 2] exactly one new vote is generated and exactly the one associated voter looses his right to vote.

Proof for L1: In the case $S_i \neq S_{i+1} \vee W_i \neq W_{i+1}$, [rule 2] had to be applied. Therefore, there exists $s \in S_{i+1}$ for which holds: $S_i = S_{i+1} \setminus \{s\}$: thus, s is the only element in $S_{i+1} \setminus S_i$. Therefore, the first part of the lemma is proven. Moreover, according to [rule 2] the following statement holds for this $s : voter(s) \in W_i$ with $W_{i+1} = W_i \setminus \{voter(s)\}$. Thus, $voter(s)$ is the only element in $W_i \setminus W_{i+1}$. Therefore, the second part of the lemma is proven.

q.e.d. (L1)

[20] The distinguish between [rule 1] and [rule 2] is not necessary for the modelling of the identified security objectives but for further work to model additional properties like vote canceling.

L2: $W_{total} = W_0 \supseteq W_1 \supseteq W_2 \supseteq ... \supseteq W_i$

Interpretation: The set of eligible voters can only decrease.

Proof for L2: This lemma is a trivial consequence of [rule 2].

q.e.d. (L2)

L3: $\forall s \in S_i : \exists j < i : voter(s) \in W_j \setminus W_i$

Interpretation: For each vote stored in the e-ballot box, there exists a voting right discarded earlier.

Proof for L3: Application of proof by induction over i, starting with $i = 1$:

Induction Base: For $i = 1$: choose $j = 0$, then this case is equal to the special case of L1 with S_1 and S_0.

Induction Hypothesis: L3 holds for some $i \geq 0$

Induction Step: For $i + 1$ holds: $\forall s \in S_{i+1}$ does either hold $s \in S_{i+1} \cap S_i$ or $s \in S_{i+1} \setminus S_i$. In the first case the statement is true according to the induction hypothesis. In the second case, L1 proves the statement.

q.e.d. (L3)

These lemmas are now used to prove the theorem.

Induction Base: All three secure state properties do hold for the initial state Z_0 because S_0 and $W_{total} \setminus W_0$ are equal to the empty set.

Induction Hypothesis: The secure state property holds for some state Z_i with $i \geq 0$.

Induction Step: It needs to be shown that for all possible states Z_{i+1} reachable by permitted state transitions from Z_i holds that a secure state is reached:

- [rule 1] $W_i = W_{i+1} \wedge S_i = S_{i+1}$; thus, $Z_i = Z_{i+1}$. Therefore, applying the induction hypothesis it holds that also Z_{i+1} is a secure state.
- [rule 2] $\exists s \in S_{i+1} : (voter(s) \in W_i \wedge W_{i+1} = W_i \setminus \{voter(s)\} \wedge S_i = S_{i+1} \setminus \{s\})$. Each of the three properties of a secure state is proven separately:
  - O.T.IneligVotes:

    Induction Hypothesis: For some $i \geq 0$ holds: $\forall s \in S_i : voter(s) \in W_{total}$

    Induction Step: Then for $i + 1$ holds: $\forall s \in S_{i+1} : s \in S_{i+1} \cap S_i \vee s \in S_{i+1} \setminus S_i$.
    - Case $[s \in S_{i+1} \cap S_i]$: this holds because of the induction hypothesis.
    - Case $[s \in S_{i+1} \setminus S_i]$: according to L1 holds: $W_i \setminus W_{i+1} = \{voter(s)\} \Rightarrow voter(s) \in W_i$ and according to L2 holds: $W_i \subseteq W_{total}$ hence $voter(s) \in W_{total}$.

    q.e.d. (O.T.IneligVotes)
  - O.T.OneVoterOneVote:

Induction Hypothesis: For some $i \geq 0$ holds: $\forall s, s' \in S_i : voter(s) = voter(s') \Rightarrow s = s'$

Induction Step: Then for $i+1$ holds: for all s and s', only the following three possibilities exist:

· Case $[s, s' \in S_{i+1} \cap S_i]$: this holds because of the induction hypothesis.

· Case $[s, s' \in S_{i+1} \setminus S_i]$: according to L1 holds: $S_{i+1} \setminus S_i = \{s\} \Rightarrow s = s'$

· Case $[s \in S_{i+1} \setminus S_i \;\wedge\; s' \in S_i]$: according to L1 holds: $W_i \setminus W_{i+1} = \{voter(s)\} \Rightarrow voter(s) \in W_i \setminus W_{i+1}$ and according to L3 holds $\exists j < i : voter(s') \in W_j \setminus W_i$.
Thus, $voter(s) \in W_i$ and $voter(s') \notin W_i$. Thus, both values can never be equal. Thus, the statement holds also in this third case.

q.e.d. (O.T.OneVoterOneVote)

– O.OSP.VoteRight:

Induction Hypothesis: For some $i \geq 0$ holds: $\forall x \in W_{total} \setminus W_i : \exists s \in S_i : voter(s) = x$

Induction Step: Then for $i+1$ holds: for $x \in W_{total} \setminus W_{i+1}$, x must be in one of the following sets:

· Case $[x \in (W_{total} \setminus W_{i+1}) \cap (W_{total} \setminus W_i)]$: this holds because of the induction hypothesis.

· Case $[x \in (W_{total} \setminus W_{i+1}) \setminus (W_{total} \setminus W_i)]$: according to L2 holds: $W_{total} \supseteq W_i \supseteq W_{i+1}$. Thus, $(W_{total} \setminus W_{i+1}) \setminus (W_{total} \setminus W_i) = W_i \setminus W_{i+1}$; thus, $x \in W_i \setminus W_{i+1}$; in addition it holds: $W_i \neq W_{i+1}$. According to L1 holds $W_i \setminus W_{i+1} = \{voter(s)\}$ for $s \in S_{i+1} \setminus S_i$. Then, deduced from $x \in W_i \setminus W_{i+1}$ it holds: $voter(s) = x$; this completes the proof for $i+1$.

q.e.d. (O.OSP.VoteRight)

q.e.d. (Security Theorem)

With this proof, it is shown that the defined secure states and allowed state transitions build a formal IT security model.

7.5 Summary

This chapter proposes the application of the Common Criteria (CC) [35] (Version 3.1) and the corresponding Common Evaluation Methodology (CEM) [36] for the evaluation of remote electronic voting systems according to the identified security, functional, and assurance requirements. The Common Criteria is an international accepted evaluation standard (ISO 15408) that strictly guides the evaluator with the Common Evaluation Methodology and that is flexible with respect to different trust models and different evaluation depths. Defining a trust model is one important part of the CC. Assumptions about the environment are defined in the security problem definition and the intruder's technical capabilities are represented in the security assurance requirements (until EAL4 an enhanced-basic attacker potential is assumed, then for EAL5 a moderate, and for EAL6 and 7 a high attacker potential is assumed). Therefore, the application of the Common Criteria overcomes the identified vulnerabilities of existing requirement documents.

In order to be able to apply the Common Criteria, the standard is introduced in Sect. 7.1. The main items are explained, namely security problem definition (including threats, organisational security policies, and assumptions), security objectives, security functional requirements (SFR), security assurance requirements (SAR), evaluation assurance levels (EAL), Protection Profiles, Security Targets, and the Common Methodolgy for IT Security Evalution. Moreover, in this section the wording from previous sections is translated to the Common Criteria syntax and semantics.

Section 7.2 concentrates on trust models for remote electronic voting systems; first in general and then concerning two examples and their implications for possible implementations: the temporary unlimited secrecy of the vote and the trustworthiness of the vote-casting device. This discussion shows how important the proper security problem definition is and that for some strong security problem definition, currently, no systems exists that would pass an evaluation.

The evaluation depth is addressed in Sect. 7.3. The assurance requirements from Sect. 6.5 are translated to CC security assurance requirements. This mapping results in the evaluation level EAL 4+.

In Sect. 7.4 evaluation according to EAL5, EAL6, and EAL7 are addressed as the Common Criteria is generally flexible with respect to the evaluation depth, but there are still a couple of open questions and research tasks to solve (not only concerning remote electronic voting but in general). It is shown that it is necessary to specify an IT security model to apply EAL6 or 7. Therefore, this section specifies such a model for some of the defined requirements for remote electronic voting systems. The correctness of this model is proven.

8

Core Protection Profile

In Chap. 6, a list of requirements for remote electronic voting systems is provided as well as the exact definition of the addressed target of evaluation. Moreover, in Chap. 7, the Common Criteria is identified as an appropriate evaluation technique for remote electronic voting systems.

In this chapter, the first step to apply the Common Criteria for remote electronic voting systems is taken by bridging from the book's syntax to the Common Criteria vocabulary, by discussing different trust models, and by mapping the assurance requirements from Sect. 6.5 to a Common Criteria evaluation assurance level.

Based on these findings, a Protection Profile needs to be developed in order to provide a basis for standardised evaluations with comparable results. In general, bridging to the Common Criteria syntax shows that much of the content of a Protection Profile already exists:

- The security problem definition.
- A corresponding list of security objectives for the TOE (while those for the operational environment do not exist because of the empty list of assumptions).
- The security objective rationale already (1:1 mapping).
- The security assurance requirements.

Such a Protection Profile has been developed in a cooperative project between the Gesellschaft für Informatik (GI - the German society of computer scientists), the Bundesamt für Sicherheit in der Informationstechnik (BSI - Federal Office for Security in Information Technology), and the Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI -German Research Center of Artificial Intelligence). This Protection Profile aims to define core requirements for remote electronic voting systems (see [161] for the certified version[1]).

[1] The PP authors completed their work in late February 2008. While writing this part of the book, version 0.27 of the Protection Profile was the actual version.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 149–174, 2009.

The work on this book and the GI/BSI/DFKI projects have run in parallel. Therefore, not all of the results from the previous chapters are included in the formal PP document [161], even though the author of this book is one of the two authors of the GI/BSI/DFKI Protection Profile. As, this GI/BSI/DFKI Protection Profile has already been published, this chapter does not provide a new Protection Profile but instead

- describes the GI/BSI/DFKI project's background, history, and discussions,
- explains why it is useful first to define a Protection Profile defining *core* requirements for remote electronic voting systems and then extend this core framework step by step,
- summarise the content of the Protection Profile,
- explains the main decisions made in the Protection Profile: the decision for lowest acceptable evaluation depth as well as a maximised acceptable trust model meaning the maximum set of assumptions and the minimum intruder's capability,
- serves as a censorious dispute with the existing Protection Profile based on the experience and knowledge from the previous chapters, and
- suggests improvements mainly with respect to the security problem definition (which has consequences for the security objective definition and the security functional requirements by definition).

Therefore, this chapter helps to link both works and to better understand the GI/BSI/DFKI Protection Profile.

8.1 Background, History, Motivation, and Discussions

Background and History. In Germany, the Gesellschaft für Informatik (GI - the German society of computer scientists) decided in 2004 to introduce remote electronic voting as the main channel for elections of their board of chairpersons. Based on the standards available at that time [37, 62, 165], they developed their own catalogue [113] in 2005 (for more information about the catalogue see Sect. 3.2.3). In parallel with this development, the GI established an expert group and gave them the task to develop a Common Criteria Protection Profile for remote electronic voting based on the same standards that were used for the GI catalogue in [113]. Participants in this expert group included employees from the Bundesamt für Sicherheit in der Informationstechnik (BSI - Federal Office for Security in Information Technology), universities, ministries, and administrations, as well as product developers and data protection authorities. Additionally, the BSI contracted the Protection

This version was under evaluation. Since June 2008 the GI/BSI/DFKI Protection Profile has been certified and published. This book is therefore based on the final and thus certified version.

Profile to the Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI -German Research Center of Artificial Intelligence) to formally develop a corresponding Protection Profile in conjunction with the expert group. A report on this work is provided in [55], and the first results are published in [155] and [154].

Motivation to Define Core Requirements. When starting the project, the aim was not to develop core requirements at first glance but to develop requirements for low level elections and particularly for the GI elections. During the discussions amongst the expert group, all members ultimately agreed on parts of the security problem definition, but they did not agree on the entire security problem definition. Some members of the expert group thought that the security problem definition was too weak because of too many assumptions about the environment, while for others, the security problem definition was too strong because they felt that too less assumptions were made about the environment. Thus, some of the experts began thinking about developing their own Protection Profile for their particular election level (for instance, staff and council work elections) that was independent from the GI/BSI/DFKI Protection Profile. Therefore, the expert group decided to distance themselves from the idea of developing a Protection Profile for low level elections and, in particular, for GI elections and decided to define the minimum set of requirements that each remote electronic voting system needs to ensure. Such a Protection Profile should serve as a basis for all other PPs for any remote electronic voting system. In particular, such a core PP can be extended by removing assumptions about the environment, thereby demanding more security functionality by the TOE. By doing so, confusion for the responsible election authorities due to the existence of various PPs that do not reference each other can be prevented. To provide an international basis, a decision was made to translate the original German Protection Profile text into English [160].

Discussions. The development of the Protection Profile was driven by regular feedback sessions with the expert group. Major issues have included the following aspects:

- *The title:* The group debated whether to include or exclude the type of addressed elections (for instance, elections in societies or universities).
  A decision was reached to exclude this information in the title. The title is Basic Set of Security Requirements for Online Voting Products, while 'Online Voting Product' in the PP is used similar to the term 'remote electronic voting system' (the German title is "Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte").
- *Demonstrable versus strict conformance:* The Common Criteria requires a decision of whether the Protection Profile needs to follow demonstrable or strict conformance.
  The final decision required strict conformance. The motivation for this decision is presented in Sect. 8.2.2.

- *Threats versus assumption:* The group needed to determine which threats should be specified as threats against the remote electronic voting system(and thereby creating security objectives that must be covered by the remote electronic voting system) and which threats can be shifted into assumptions and corresponding security objectives to the environment. The assumption concerning the trustworthiness of the vote-casting device was the predominate topic of discussion.
  The content of the security problem definition including a list of threats and a list of assumptions is provided in Sect. 8.2.3.
- *The evaluation assurance level:* As the PP addresses core requirements and intends to serve as a basis for any kind of remote electronic voting system evaluation, the task was to define the minor acceptable evaluation assurance level. Developers prefer to decide for a lower level, while voters and security experts tend to argue for a higher level in order to provide more confidence in the evaluation and thus the system.
  A decision was reached to demand EAL2+. The motivation for this decision is presented in Sect. 8.2.5.
- *The point in time for voter authentication:* The voter could be identified and authorised either before the ballot is displayed on his vote-casting device or just before the vote cast.
  The group decided to leave this open and allow both possibilities.
- *Observation:* The question was whether to include or exclude election observation.
  A decision was reached to exclude observation as the PP intent to address basic requirements. However, observation can be extended correspondingly if a responsible election authority decides to include it.
- *Spoilt votes:* Discussion also involved whether to include or exclude requirements demanding the possibility to explicitly spoil votes and/or a warning if voters try to spoil their votes.
  It was decided not to include such requirements with the justification that the Protection Profile can be extended if necessary.
- *Error handling:* The group also needed to determine how to handle which kind of errors and problems that occur during the polling phase.
  This is still an open issue (see Sect. 8.3).

In addition to these discussions in the expert group, a parallel discussion proceeded between the German Federal Ministry of the Interior and the BSI. They demanded the following extension on the first page:

"The compliance to the requirements specified in this Protection Profile is sufficient to securely implement some kinds of elections in associations, for boards and bodies such as at universities, within in the scope of education and research, and in particular other non-political elections with low attack potential. In order to securely perform online elections with a higher attack potential, like works council elections or parliamentary elections, further security requirements are to be defined, and they are to be enforced by demonstrable measures satisfying the assumptions to the application environment as they are described here. More advanced requirements satisfying the assumptions to the voting environment with higher risk of attacks can seamlessly be built upon the core of the central requirements described here, and they may supplement this core but, under any circumstances, shall not replace it."

8.2 The GI/BSI/DFKI Protection Profile

Knowing the background, history, motivation, and discussions related to the GI/BSI/DFKI Protection Profile ([161] and the English version [160]), a section by section review of the certified PP document is presented in this subsection. Thereby, the vocabulary provided in the glossary of this book is used and if the PP uses a different one, the mapping is provided in a corresponding footnote and in Table C.2 in the appendix.

8.2.1 Introduction/TOE Overview

The GI/BSI/DFKI Protection Profile contains a very detailed discussion of the TOE in section 1. Before section 1.1, in the section prefix, information about the motivation to develop such a *core* Protection Profile for remote electronic voting is given. This prefix also describes those types of elections according to the authors for which these core requirements might be enough and how to extend this Protection Profile in case the security problem definition and consequently the trust model for a particular election differs from the one defined in the GI/BSI/DFKI Protection Profile.

This part also states that the decision for the 'extension of the security problem definition or not' has to be made by the responsible election authority. In addition, the list of existing catalogues on which this Protection Profile is based is provided: the 'Recommendation of the Council of Europe' [37], the PTB catalogue addressing 'Online-Voting Systems for Non-parliamentary Elections' [62], and the GI catalogue for 'Internet-based Elections in Societies' [113] (these catalogues are discussed in Sect. 3.2).

Section 1.1 (PP Reference) specifies the references, containing title, publisher, authors, version number, registration number, and key words.

Section 1.2 (TOE Overview) contains the following subsections[2]:

- TOE Type (1.2.1): This subsection restricts the TOE functionalities to those for the polling phase and tallying phase, while the election phase transition from the election setup phase to the polling phase is handled by a corresponding assumption that the system is properly installed and

[2] The CC recommends to further discuss in section 1.2 'Usage and major security features of a TOE' (here 1.2.4), 'TOE Type' (here 1.2.1), and 'Available non-TOE hardware/software/firmware' (here 1.2.5).

configured. The tasks of the TOE and, in particular, the server-side TOE and the client-side TOE are described in this subsection.
- Abbreviations and Glossary (1.2.2): Abbreviations are presented for CC specific vocabulary, and a glossary for electronic voting terminology is included.
- General Security Expectations of the TOE (1.2.3): This subsection technically interprets the election principles.
- Usage and Important Security Characteristics (1.2.4): This subsection discusses the following aspects: the status of the TOE prior to the polling phase including the tallying phase (1.2.4.1), the process description for the polling phase in general and for the voting process in particular (1.2.4.2), the description of the system state after the tallying phase (1.2.4.3), the proposal for the operations by the voter (1.2.4.4) and by the poll workers[3] (1.2.4.5), the discussion of malfunctions, self-test and recovery (1.2.4.6), and the auditing mechanisms (1.2.4.7).
- Required non-TOE hardware/firmware/software (1.2.3)

Moreover, at all fifteen application notes are defined in section 1 of the GI/BSI/DFKI Protection Profile. Most of them address the Security Target author; for instance, that he is allowed to add security objectives by removing assumptions or adding functionality (see application note 1, 2, 4, 10, 13). Also, particularly in application notes 5, 6, and 12, directions are repeated, stating that the new security objectives must not contradict with the existing ones. Other application notes address refinements that must be made by the Security Target author (for instance, the functionality distribution between the client-side TOE and the server-side TOE in application note 3, the concrete description of the voting process in application note 9, the list of allowed vote-casting devices in application note 14, and the list of necessary hardware, firmware, and software in application note 15). The remaining application notes (for instance, 7, 8, and 11) provide additional explanations, such as a description with respect to possible technical implementations to ensure the secrecy of the vote.

8.2.2 Conformance Claims

In the BSI/GI/DFKI Protection Profile, section 2 includes the following:

- The PP is compliant with the Common Criteria version 3.1 (for both parts 2 and 3) as this is the newest version[4].
- The PP demands an augmented EAL2 with ALC_ CMC.3, ALC_ CMS.3, ALC_ DVS.1, and ALC_ LCD.1[5].

[3] poll workers are called election officers in the GI/BSI/DFKI PP.

[4] When the project started, V3.1 was not yet available. Thus, the first drafts were based on V2.3.

[5] The decision for EAL2+ is further discussed in Sect. 8.2.5.

- The PP demands strict conformance.
- The PP is not based on existing PPs (by virtue of removing the topic ‘PP conforms to other PPs’).

As stated in Sect. 8.1, one of the main issues for discussion was the decision to require strict conformance. Therefore, this part of the conformance claim is reviewed in more detail. The Common Criteria distinguishes between strict and demonstrable conformance. This conformance claim addresses the required relationship between a Protection Profile and a Security Target (ST) that tries to conform to a particular PP as part of a TOE evaluation. The distinction between strict and demonstrable conformance was only recently introduced with the Common Criteria, Version 3.1. Thus, the approach is rather new. The concept of a PP rationale to show the compliance of a ST to the PP as presented in previous CC versions does not exist anymore. It has been replaced by the concept of a conformance claim rational, which exists in two different shapes. Strict conformance is a more rigorous conformance claim than the PP rationale, while demonstrable conformance is a weaker claim than the PP rationale.

However, only applying one of the conformance claims for the whole Protection Profile is possible, thereby not allowing assignment of different conformance claims to separate parts of the Protection Profile. According to the Common Criteria, the differences of both claims are defined in the following way:

- Strict conformance: A very strict relationship exists between the PP and the ST. ”Strict conformance is oriented to the PP-author who requires evidence that the requirements in the PP are met, that the ST is an instantiation of the PP,[...] the ST specifies that the TOE does at least the same as in the PP, [...]”. ”Strict conformance is expected to be used for stringent requirements that are to be adhered to in a single manner”.
- Demonstrable conformance: ”The PP and the ST may contain entirely different statements that discuss different entities, use different concepts etc. However, the ST shall contain a rationale on why the ST is considered to be ”equivalent or more restrictive” than the PP [...]. Demonstrable conformance allows a PP author to describe a common security problem to be solved and provide generic guidelines to the requirements necessary for its resolution, in the knowledge that there is likely to be more than one way of specifying a resolution.” ”Demonstrable conformance is orientated to the PP-author who requires evidence that the ST is a suitable solution to the generic security problem described in the PP.”
  In particular, this means the following for the definition of SFRs and SARs in the ST:
  SFRs: ”The conformance rationale in the ST shall demonstrate that the SFRs in the ST are equivalent (or more restrictive) than the SFRs in the PP. This means that all TOEs that would meet the SFRs in the ST would also meet the SFRs in the PP [...]”

> SARs: "The ST shall contain all SARs in the PP, [...]" [35].

When discussing the conformance claim for the GI/BSI/DFKI Protection Profile, but also in general, one can argue for and against both conformance types.

- The antagonists of demonstrable conformance argue that unsuitable solutions will also be evaluated against the PP if the ST author can argue well. As it is a new approach, no one really knows how large the gap between the PP and ST will ultimately be. Therefore, from their point of view, gaining experience with what is really possible, the constraints, and the limits with demonstrable conformance is necessary. Further, they argue that since remote electronic voting is a controversial and political topic, it should not be approached with the demonstrable conformance proposition.
- The supporters of demonstrable conformance argue that so many different approaches exist for the implementation of a remote electronic voting system, which contradicts the idea of strict conformance.Moreover, deciding whether all existing 'secure' approaches can possibly comply with the GI/BSI/DFKI Protection Profile is not realistic if it requires strict conformance. An additional problem in Germany is that the BSI is not allowed to certify remote electronic voting systems that do not conform to the GI/BSI/DFKI PP according to a directive from the Federal Ministry of the Interior[6]. This directive supports the intention that all remote electronic voting systems shall at least ensure these core requirements, but it creates problems if some fail because of the strict conformance demand. The idea to later downgrade from strict to demonstrable conformance if such a situation appears might help the developers but will lead to heated debates and negative publicity.

When discussing the conformance claim for the GI/BSI/DFKI Protection Profile and also in general, one can argue for and against both conformance types. Generally, a mix of both approaches would be preferable. For instance, strict conformance could be required for the security problem definition and the security assurance requirements as a fundamental discussion base, while demonstrable conformance could be allowed for the security functional requirements to cover all possible approaches. The CC community is attempting to establish such an approach, but this work is currently still in draft form.

8.2.3 Security Problem Definition

In section 3, the GI/BSI/DFKI Protection Profile first defines a set of considered assets and subjects. These items are used later in the definition of threats,

[6] If one system is not conform, a special application to the Federal Ministry of the Interior is required to start/continue the certification process. The above information is according to an email from February 2008 (by M. W.).

organisational security policies, and assumptions. Asserts are either considered as user data or TSF data[7]. According to the GI/BSI/DFKI Protection Profile, the following assets have to be protected by a TOE:

- Authentication message (user data)
- Authentication data (TSF data)
- Identification data (user and TSF data)
- Ballot data (user data)
- Ballot (user data)
- Vote (user data)
- E-vote[8] (user data)
- Confirmation[9] (user data)
- Election data (user data)
- Polling phase data (user data)
- Audit records (user data)
- Result (user data)

Subjects are either intended users (voters or poll workers) or attackers. The PP distinguishes between the following types of attackers:

- Network attacker
- Voter[10]
- Ineligible voter
- Person who has access to data stored in the TOE after the tallying phase

With respect to the attackers, the GI/BSI/DFKI Protection Profile defines the properties of each attacker type, including the capability and the available equipment. In addition, the differences between direct and indirect methods as well as between active and passive opportunities to attack the TOE are explained (see subsection 3.1.1 of the PP).

GI/BSI/DFKI - Protection Profile Threats

The following threats are defined in section 3.1.2:

- T.UnauthorisedVoter: An ineligible voter[11] or an elector[12] cast a vote .
- T.Proof: An elector uses data on his vote-casting device that are produced by TOE during the polling phase to prove to a third party that he has voted in a certain way.

[7] According to the Common Criteria user data is “data created by and for the user, that does not affect the operation of the TSF” [35]. Correspondingly, TSF data is “data created by and for the TOE, that might affect the operation of the TOE” [35].

[8] The PP uses ‘vote record’ instead of ‘e-vote’.

[9] The PP uses ‘acknowledgement’ instead of ‘confirmation’.

[10] The PP uses ‘registered voter’ instead of ‘voter’.

[11] The PP uses ‘unauthorised voter’ instead of ‘ineligible voter’.

[12] The PP uses ‘voter without the right to vote’ instead of ‘elector’.

- T.IntegrityMessage: A network attacker intervenes directly in the network in order covertly to delete, generate, replay, or modify data while it is in transmission.
- T.SecretMessage: A network attacker intervenes directly in the network in order to sniff messages relating to the polling phase while they are in transmission.
- T.AuthenticityServer: A network attacker redirects voters to a malicious voting server. A network attacker redirects the voter to a hoax (voting) server. Hence, the voter does not communicate with the authentic voting server[13].
- T.ArchivingIntegrity: A person who has access to the data stored by the TOE after the tallying phase, forges or modifies the stored election result, the stored election data, and, if necessary, the audit records or further data in order to influence the result of any recount or changes the stored result.
- T.ArchivingSecrecyOfVoting: A person who has access to the data stored on the TOE after the tallying phase and where applicable has supplementary data, e.g. decryption keys [..], can link a voter to his vote on the basis of the data stored in the TOE (in plain text or encrypted form).

For each of these threats, the PP provides an attack description, including a statement regarding the motivation, the applied methods, the opportunity, the used vulnerability, and the attacked assets.

GI/BSI/DFKI - Protection Profile OSPs

In section 3.2 of the Protection Profile, the following fifteen organisational security policies are defined:

- P.Abort: The voter shall be able to cancel[14] the voting process[15] at any time prior to the vote casting without losing his right to vote.
- P.EndingElection: The inadvertent ending of the polling phase ahead of time shall be prevented. However, the poll workers[16] are able to end of the polling phase before the planned ending time of election.
- P.EndofElection: After the end of the polling phase, it is no longer possible to open or continue a new voting process; in particular it is no longer possible to cast a vote.
- P.SecrecyOfVotingElectionOfficer: The responsible election authoritys are not in the position to use the TOE to breach the secrecy of the vote[17] during the polling phase.

[13] The PP uses ‘election server’ instead of ‘voting server’.

[14] The PP uses the term ‘abort’ instead of ‘cancel’.

[15] The PP uses the term ‘polling process’ instead of ‘voting process’.

[16] The PP uses the term ‘election officers’ instead of ‘poll worker’.

[17] The PP uses the term ‘secrecy of voting’ instead of ‘secrecy of the vote’.

- P.IntegrityElectionOfficers: The poll workers are not in the position to put e-votes into the e-ballot box[18] using the TOE. Neither are they able to delete e-votes already in the e-ballot box or selectively to modify them. In particular, no functions exist that allow the poll workers to reset the TOE to its initial state after the polling phase has begun.
- P.IntermediateResult: It shall be ensured that the poll workers are not able to compute intermediate results.
- P.OverhasteProtection: The TOE shall only store e-votes in the e-ballot box that the voter has finally cast after an explicit double-check.
- P.Correction: No limit shall be placed on the number of times that the voter can correct his vote before finally casting it. He shall also be able to correct it after an explicit double-check has been performed.
- P.Acknowledgement: The elector receives a corresponding confirmation regarding the permission or refusal and the success or failure of his vote casting.
- P.Failure: At the initial start-up and at request the poll workers shall be able to recognize, by performing a self-test at the server-sided TOE, when a failure of the integrity of the TOE security functionality (TSF) or the user and TSF data, occurs which endanger the proper operation of the TOE. After a crash / shutdown of the server-sided TOE, of the voting server or a communication failure or a failure of the storage media, the poll workers shall be able to restart the polling phase. Here, the TOE shall ensure the integrity of the polling phase data.
- P.Audit: The server-sided TOE shall at least audit the events listed in section 1.2.4.7 [of the PP] including the time of the event's occurrence and shall store the audit records in the IT environment of the server-sided TOE in a way that they are protected against unauthorized manipulations. The poll workers shall be able to review them.
- P.OneVoterOneVote:It shall be ensured that each voter can only cast one vote and that no voter unjustly loses his right to vote. This shall be ensured especially in the case of aborts of the voting process caused by the voter, the client-sided TOE, the IT environment of the TOE or the network; and this shall be ensured for any restart of the polling phase.
- P.AuthElectionOfficer: The TOE shall identify and authenticate the poll workers prior to every other action. The authentication function of the TOE shall be such that it supports a separation of duty within the group of poll workers. The operations to start, restart and end the polling phase, as well as to start the tallying with determination of the election result can only be carried out once they have been authorised by a minimum of two authenticated poll workers.
- P.StartTallying: The poll workers can not start the tallying until the polling phase has been ended.
- P.Tallying: All e-votes stored in the e-ballot box after the end of the polling phase are correctly evaluated and are fed into the election results.

[18] The PP uses the term 'ballot box' instead of 'e-ballot box'.

GI/BSI/DFKI - Protection Profile Assumptions

The Protection Profile also defines a few assumptions to the environment[19]. These are provided in this paragraph . Since these assumptions were a major focus in the discussions with the advisory board, a justification for each assumption is provided:

- A.ElectionOfficers: The poll workers access the user and TSF data only by the application of the server-sided TOE functionality. The poll workers are sufficiently trained in order to understand the secure operation of the TOE and use the TOE in the intended way. Each individual poll worker received his identification data and authentication measure, and does not forward this data to other persons. By choosing the poll workers, the responsible election authority shall ensure that it is not possible for one single person to obtain admission and access to the server-sided TOE.
  Justification: Two reasons justify this assumption. First of all, poll workers have already been proven trustworthy through their work in traditional elections (see also Sect. 10.1). Second, If this assumption is removed, the voting server, including the hardware and the operating system, must be included in the TOE since the software alone cannot prevent attacks by inside intruders. Thus, the evaluation would become much more complex, expensive, and time-intensive.
- A.VoteCastingDevice: The voter acts responsibly in securing the vote-casting device. It is assumed that each voter that installs or uses the client-sided TOE does so in such a way that the vote-casting device can neither observe nor influence the voting process. This includes the assumption that the voter does not manipulate his vote-casting device on purpose. The vote-casting device is able to properly display the ballot, to properly transfer the voter's input to the voting server and to delete the e-vote after the polling process.
  Justification: This assumption corresponds to A.TamperClient from Sect. 7.2.1 in which the consequence of not including this assumptions is discussed. Based on this analysis, a decision was reached to include the trustworthy vote-casting device problem in an assumption.
- A.ElectionServer: Protection of the voting server against attacks that originate from the insecure network is provided by the application of a security concept for the network connection, which prevents access to the election server from network attackers.
  Justification: A couple of tools and techniques already exist to ensure the implementation of a secure voting server. By removing this assumption, all of these tools must be included in the TOE. Thus, the evaluation would become much more complex, expensive, and time-intensive.

[19] This section does not discuss the assumptions A.ElectionPreparation, E.Observation, and A.AuthData because these assumptions are addressed in Sect. 6.2.

- A.Availability: The robustness, the quality of service and the availability of the network and of the voting server are assumed.
  Justification: Clearly, Denial of Service attacks are a serious threat for all remote electronic voting systems. However, the polling phase could be designed to last for several days, even weeks, which would make Denial of Service attacks unattractive. In addition, redundant voting servers could be implemented but should not be part of the TOE because of the consequences for the evaluation.
- A.ServerRoom: No-one other that the poll workers, gains entry to the server room or admission to the voting server for the duration of the polling phase and until the vote tallying.
  Justification: Since physical access can be ensured by organisational measures, this assumption can not be criticised as too hard. Note that this assumption is needed to support A.ElectionServer and A.ElectionOfficers because the voting server is not required to be tamper-resistant.
- A.SystemTime: The correct time is made available by the server's IT environment; and it corresponds to the proper time. The required exactness is defined by the responsible election authority.
  Justification: This assumption does not need to be further discussed as it can be ensured by organisational means and does not need to be part of the evaluation. However, by removing this assumption, the evaluation would again become much more complex, expensive, and time-intensive.
- A.DataStorage: The storage media is functioning correctly, that is the integrity and the availability of all stored user and TSF data is ensured. Errors during the storage of e-votes in the e-ballot box are reported to the TOE security functionality.
  Justification: There are corresponding components already available. Thus, again, by removing this assumption, the evaluation would again become much more complex, expensive, and time-intensive.
- A.AuditTrailProtection: The IT environment of the server-sided TOE stores the audit records generated by the server-sided TOE in a way that they are protected against unauthorized manipulations.
  Justification: Tools and techniques already exist to ensure this requirement in the IT environment. Nevertheless, the removal of this assumption would again result in the necessary inclusion of all of these tools in the TOE. Thus, the evaluation would become much more complex, expensive, and time-intensive.
- A.Buffer: Ballot data or e-votes buffered on the vote-casting device outside the scope of control of the TOE are not available anymore after the voting process.
  Justification: This assumption is necessary in order to enable Web browser based remote electronic voting systems, which are not able to delete data in temporary files on the vote-casting device.
- A.AuthenticityServer: The voter verifies whether he communicates with the authentic server-sided TOE.

Justification: This assumption is necessary in order to enable Web browser based remote electronic voting systems, which secure the communication only via SSL.

- A.ArchivingSecrecyOfVoting: For all additional data – such as decryption keys which would link a particular voter to his vote after the completion of the tallying process [..] – the life cycle control and access control defined by the remote electronic voting system is effectively implemented by the poll workers with appropriate technical and organizational measures.
  Justification: This assumption is necessary in order to enable systems which know and store a link between encrypted e-votes and the voter ID. Here, it is necessary to protect corresponding decryption keys.
- A.ProtectedCommunication: The IT environment facilitates a communication connection between the vote-casting device and the voting server which is protected against modifications and disclosure.
  Justification: This assumption is necessary in order to enable Web browser based remote electronic voting systems, which secure the communication only via SSL.

8.2.4 Security Objectives and Functional Requirements

In section 4.1, the Protection Profile describes a security objective for each threat and the TOE which prevents it and a security objective for each organisational security policy and the TOE which achieves it. The security objective rationale part shows that almost all security objectives for the TOE can only be reached under the conditions described in the defined assumptions for the environment. In addition to the list of security objectives for the TOE, the PP defines a list of security objectives for the environment in section 4.2. The corresponding rationale in section 4.3.3 shows the 1:1 mapping.

In section 5.1, the Protection Profile addresses the security functional requirements. However, in order to do so, the following items are first defined:

- Subjects – voter and poll workers
- Objects – authentication message, identification data, audit records, confirmation, entry in the electoral register, e-vote, vote, ballot, ballot data, election data, result, and intermediate result
- Security attributes – number of authorisations for a poll worker operation, polling period, right to vote, and voting period

Based on this items the following security functional requirements have been defined: FAU_ GEN.1, FAU_ SAR.1, FDP_ DAU.1, FDP_ IFC.1A/B, FDP_ IFF.5, FDP_SDI.2, FDP_RIP.1A/B, FDP_UCT.1, FDP_UIT.1, FIA_ATD.1, FIA_ UAU.1/2/6, FIA_ UID.1/2, FIA_ USB.1A/B, FPR_ ANO.1, FPR_ UNL.1A/B, FPT_ RCV.1/4, FPT_ TST.1, FTA_ SSL.3/4, FTA_ TSE.1, and FTP_ TRP.1. In the security functional rationale section (see PP section 5.3), the compliance of the security objectives is detailed.

8.2.5 Security Assurance Requirements

After a long discussion in the development phase of the project, a decision was made to require EAL2+ as the lowest acceptable evaluation assurance level allowed for the responsible election authority. In particular, EAL2 in combination with the following SFR components is required by the PP: ALC_ CMC.3 (substituting ALC_ CMC.2), ALC_ CMS.3 (substituting ALC_ CMS.2), ALC_ DVS.1, and ALC_ LCD.1. Accordingly, the following security assurance requirements have to be met:

- ADV: Development
  - ADV_ ARC.1 Security architecture description
  - ADV_ FSP.2 Security-enforcing functional specification
  - ADV_ TDS.1 Basic design
- AGD: Guidance documents
  - AGD_ OPE.1 Operational user guidance
  - AGD_ PRE.1 Preparative procedures
- ALC: Life-cycle support
  - ALC_ CMC.3 Authorisation controls (instead of .2 Use of a CM system)
  - ALC_ CMS.3 Implementation representation CM coverage (instead of .2 Parts of the TOE CM coverage)
  - ALC_ DEL.1 Delivery procedures
  - ALC_ DVS.1 Identification of security measures (added)
  - ALC_ LCD.1 Developer defined life-cycle model (added)
- 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.1 Evidence of coverage
  - ATE_ FUN.1 Functional testing
  - ATE_ IND.2 Independent testing - sample
- AVA: Vulnerability assessment
  - AVA_ VAN.2 Vulnerability analysis

EAL1 is not valid to ensure security for any type of election, and the augmentation of EAL2 is reasonable for the following reasons:

- EAL1 is not satisfying because
  - the system architecture is not evaluated because ADV_ ARC is missing,
  - The evaluator does not test the TOE, nor does he checks the tests that are executed by the developer because ATE_ FUN is missing, and

  - the security measures for the delivery procedure are not evaluated though essential because ALC_ DEL is missing.
- EAL2 is augmented by the addition of the following elements:
  - ALC_ CMC.3 Authorisation controls
    ALC_ CMC.3 is recommended for use instead of ALC_ CMC.2 (Use of a CM system). The two most important added elements compared to ALC_ CMC.2 are the following:
    - "The CM system shall provide measures such that only authorised changes are made to the configuration items.
    - The evidence shall demonstrate that all configuration items are being maintained under the CM system." [35]
  - ALC_ CMS.3 Implementation representation CM coverage
    ALC_ CMS.3 is recommended for use instead of ALC_ CMS.2 (Parts of the TOE CM coverage). The most important added aspect compared to .2 is that ALC_ CMS.3 requires the placement of the entire TOE under a configuration management system.
  - ALC_ DVS.1 Identification of security measures
    This component is added.
  - ALC_ LCD.1 Developer defined life-cycle model
    This component is added.

  The extensions to ALC_ CMC.3 and ALC_ CMS.3 allow the CM system to control changes to the whole TOE and ensures that the TOE is only modified in a controlled manner with proper authorisations. Therefore, even if vulnerabilities and backdoors cannot be excluded, you will know the person who is responsible in case something bad happens. EAL2, which is augmented with these components, requires the use of corresponding security mechanisms to protect the development environment, while in EAL2 itself, the user needs to trust the developer. ALC_ DVS.1 and ALC_ LCD.1 have only been added because the dependencies of ALC_ CMC.3 require it.

8.3 Comparison, Open Points, and Suggestions for Improvements

This section compares the approach provided in this book with the GI/BSI/ DFKI Protection Profile. Based on this comparison, open points in the PP are identified and suggestions for improvements are provided[20].

[20] Rössler also points out vulnerabilities of the GI/BSI/DFKI in his thesis [126] (in particular with respect to ""real" democratic political elections on a large scale and/or at a high level"). Most of the vulnerabilities he mentions are addressed in this book as well. There is one exception: Rössler proposes to add the role "observer".

8.3.1 Introduction/TOE Overview

Much of the content provided in the first section of the BSI/GI/DFKI Protection Profile is described in Sect. 6.2 and in appendix C (Table C.1 maps words from one glossary to the other one). In some aspects, the PP introduction is more detailed and discusses the TOE from different views. This is necessary as the Protection Profile must contain enough information for PP evaluators, ST authors, TOE evaluators, and the responsible election authority. These people should not need additional literature to understand the PP.

There are two important conceptual differences between Sect. 6.2 of this book and section 1 in the GI/BSI/DFKI Protection Profile:

- In Sect. 6.2, the scope of the TOE covers software, hardware, and firmware in the remote electronic voting system, while in [161] only the voting software is covered.
- The implementation of a Web browser solution as voting client technique is enabled in [161], whereas it is excluded in Sect. 6.2. This causes corresponding security objectives from Chap. 6 which define requirements for the client-side voting software are reformulated in the GI/BSI/DFKI Protection Profile to requirements for the remote electronic voting system in general.

Both differences are caused by the fact that the concept of assumptions about the environment is not applied in Sect. 6.2 of this book, while it has in the GI/BSI/DFKI Protection Profile. However, both identified differences restrictions are reasonable as [161] addresses core requirements for low level elections where corresponding assumptions about the environment are acceptable and the intruder's technical capability is low.

Open Points and Suggestions for Improvements. Despite the previous explanations for the main differences, the following improvements for section 1 of the GI/BSI/DFKI Protection Profile [161] are proposed:

- Section 1.2.1 of the GI/BSI/DFKI Protection Profile, describes that the tallying process is started at the voting server. Here, it is recommended to run the tallying process on a separate device. Such a separation might be preferable in praxis, especially if the remote electronic voting closes before the traditional polling stations, such as in Estonia. Thus e-votes can be stored securely and offline from the end of the electronic voting period till the end of the paper one.
- In accordance to the approach in this book, the GI/BSI/DFKI Protection Profile simplifies the architecture to one voting server even though the existing remote electronic voting systems implement two or more voting servers. This simplified approach is meaningful but should be explained and discussed in the introduction. By leaving such a discussion out, the reader might have the impression that having one voting server is the proposed solution while the opposite is actually the case. In this context, the importance of the separation of duty principle should also be explained.

Moreover, an application note should be added which informs the Security Target author which additional security requirements he needs to define in case different voting servers are implemented (and might also communicate with each other).

8.3.2 Conformance Claims

As in the book's approach, no statement about the conformance is made; only general open points regarding the decision for 'strict conformance' are identified, and suggestions for improvements are included.

Open Points and Suggestions for Improvements. Version 0.27 of the GI/BSI/DFKI Protection Profile requires strict conformance. Proponents argue that the functionality of the TOE can be extended by extending the security problem definition and adding corresponding security requirement components, but these naturally must not contradict with the existing ones. For the author's understanding of strict conformance, known systems cannot be evaluated because the necessary functionality extension would lead to contradicting requirements. For instance, the Estonian system (see Sect. 9.2 for a description) allows vote updating. This contradicts the rules of the security functional component FDP_ IFF.1A.

In addition, with the GI/BSI/DFKI Protection Profile demanding strict conformance, remote electronic voting systems implementing more than one voting server (in order to increase the trustworthiness), more than one e-ballot box (because more than one poll run in parallel), or more than one electoral register (because the voters are grouped depending on the voting options on their ballot) may not be compliant to the rigorous requirements. However, this would exclude most of the available remote electronic voting systems.

8.3.3 Security Problem Definition

(A) Assets, Subjects, Threat Description

Asserts as such are not discussed in the approach of this book. With respect to the subjects, the definition of the types of attackers is different in both documents. This book distinguishes between the intruder types (see Sect. 4.3): outside intruders, inside intruders, and malicious voters.

The type 'malicious voters' replaces the PP subjects 'voters' and 'ineligible voters'. 'Outside intruders' are equal to the PP subject 'network attackers'. The PP subject 'the users who have access to the data stored in the TOE after the tallying phase' constitutes only a very small subset of the group of possible 'inside intruders' in the PP. Therefore, this book discusses more possible attacks than the Protection Profile.

With respect to the description manner for the threats, small differences appear between both approaches. In Sect. 4.3 of this book, a description of the attack itself and the intruder's motivation are provided, while the other aspects of the Protection Profile description, which include the intruder's methods, the opportunities, and the attacked assets, are only indirectly treated. For instance, whenever the threat description contains '*to* do something *in order to* reach his goal', the applied method is considered 'indirect' according to the CC, while whenever the first part of this term is missing (and it remains '*in order to* reach his goal'), the applied method according to the CC is 'direct'.

Open Points and Suggestions for Improvements. The concept of inside intruder is not introduced and discussed in the GI/BSI/DFKI Protection Profile. However, section 3.3 of the PP includes a related assumption, in which poll workers are assumed to only use the TOE to access user and TSF data; that is, they only use the TOE functionality but do not bypass the server-side voting software. However, under the condition that the security problem definition must be understandable for the responsible election authority, the consequence of this assumption and the exclusion of inside intruders on the list of subjects should be made explicit. The responsible election authority must understand the consequences and whom they have to trust in order to develop corresponding organisational measures.

(B) Threats

The PP threats (see subsection 3.1.2 of the PP) can be mapped to the threats listed in Chap. 6 of this book. This mapping is illustrated in Table 8.1. In the case that one threat in the GI/BSI/DFKI Protection Profile is split into two threats in this book, both threats are named in one row. In the case that one of the threat formulations from the PP or the book is more detailed, the more detailed threat is labelled with an asterisk '*'.

Result. A couple of threats defined in Chap. 6 are not considered in the GI/BSI/DFKI Protection Profile. These are: (caused by an inside intruder) T.UnauthVotes, T.Tamper-ServerB, T.ElectionSecrecyB, T.IntegVotes,

Table 8.1. Mapping threats from [160] to those from Sect. 6.3

GI/BSI/DFKI PP	Book
T.UnauthorisedVoter	T.InneligVoter, T.OneVoterOneVote
T.Proof	T.ProofGenA*, T.ProofGenB*
T.IntegrityMessage*	T.DeleteMsgNet, T.AlterMsgNet
T.SecretMessage	T.ElectionSecrecyNet, T.IntResultNet
T.AuthenticityServer	T.WrongServer
T.ArchivingIntegrity	T.IntegElecData
T.ArchivingSecrecyOfVoting	T.ElectionSecrecyA

T.AffectCounting, and (caused by an outside intruder) T.AC, T.PersonalDataNet, T.DeleteRecord, T.SecrectAuthNet, T.WrongSW, T.TamperServerB, and T.TamperClient. However, the next sections show that some threats are shifted to organisational security policies while others are subsumed by corresponding assumptions about the environment.

Open Points and Suggestions for Improvements. Two main open points and suggestions for improvements can be identified as a result of the mapping.

- Table 8.1 illustrates that T.UnauthorisedVoter is mapped to two threats, namely T.Inelig-Voter and T.OneVoterOneVote. Both threats address different types of intruders. The former case refers to an ineligible voter, and the second case refers to a malicious elector. Both types are named in the threat T.UnauthorisedVoter, but since both types of attacks require different countermeasures, a differentiation of these cases that are already listed within the threat definitions is reasonable.
- The next recommendation is more crucial. The formulation of T.Proof is very weak. The type of data that the voter can use from his vote-casting device in order generate proof is not clearly defined. Nevertheless, a decision for T.ProofGenB has dramatic consequences in the security functions of a remote electronic voting system that is able to prevent the threat (see the discussion in Sect. 7.2.1).

(C) Organisational Security Policies

According to the mapping of the threats in both documents in this section, the list of organisational security policies from the GI/BSI/DFKI Protection Profile (listed in section 3.2) is compared to the list of functional requirements that are defined in chapter 6. The comparison is illustrated in Table 8.2.

Result. According to the threat comparison, some of the organisational security policies that are defined in Chap. 6 are not considered in the GI/BSI/DFKI Protection Profile. These are: P.DataLoss, P.Feedback, P.NoInteract, P.Availability, P.StoreAllVotes, P.AccurRep, P.AccurDisp, P.EqualRep, P.Transmission, P.SecrecyAfterBreakd, and all OSPs labelled as non-core namely P.InfoPW, P.ClosePW, P.AdequNoVotes, P.Spoilwarning, P.Spoil, P.AdequatNoBallotOpt, and P.CompatClient. However, the next section shows that P.Availability and P.DataLoss are subsumed by corresponding assumptions about the environment.

Open Points and Suggestions for Improvements. One main improvement can be identified as a result of the mapping. Some organisational security policies in the GI/BSI/DFKI Protection Profile contain more than one different policy. To make it more readable and more easily verifiable in the rationale section, those OPSs should be split to separate ones.

Table 8.2. Mapping OSPs from [160] to those from Sect. 6.4

GI/BSI/DFKI PP	Book
P.Abort	P.Interface
P.EndingElection	P.PWClosePoll
P.EndOfElection	P.PWInterface
P.SecrecyOfVotingElectionOfficer	P.AvailInfo
P.IntegrityElectionOfficer	P.PWInterface
P.IntermediateResult	P.PWInterface
P.OverhasteProtection	P.Interface
P.Correction	P.Interface
P.Acknowledgement	P.Confirmation
P.Failure	P.SelfCheck, P.ErrorRecovery, P.PWInterface
P.Audit	P.Auditing, P.PWInterface
P.OneVoterOneVote	P.VoteRight, P.VoteRightExc
P.AuthElectionOfficer	P.SepDuty, **T.AC**
P.StartTallying	P.PWInterface
P.Tallying	P.AccurCalc

(D) Assumptions

In this section, the list of assumptions in the GI/BSI/DFKI Protection Profile is matched to a security problem definition from this book. This mapping is illustrated in Table 8.3.

Result. In this mapping, different types are represented in the right column. These types include the following ones:

- Assumptions from Sect. 6.2
- Most of the remaining threats from Sect. 6.3 caused by an inside or outside intruder
- Two of the remaining organisational security policies from Sect. 6.4
- Two of the organisational security policies addressing the audit system (see Sect. 6.4.3)
- Three PP assumptions could not be mapped to the book's items namely A.AuthenticityServer, A.ArchivingSecrecyOfVoting, and A.Protected-Communication. The first two assumptions address Web browser solutions.SSL can be used to ensure the communication with these assumptions. The third assumption is needed for those systems which endanger the election secrecy if corresponding decryption keys become available.

Table 8.3. Mapping assumptions from [160] to aspects in Sect. 6

GI/BSI/DFKI PP	Book
A.ElectionPreparation	A.ProperConfig
A.Observation	A.ProtectedEnvironment
A.ElectionOfficers	T.UnauthVotes, T.TamperServerB, T.ElectionSecrecyB, T.AffectCounting
A.AuthData	A.AuthToken
A.VoteCastingDevice	T.TamperClient
A.ElectionServer	T.TamperServerA
A.Availability	P.Availability
A.ServerRoom	T.AC
A.DataStorage	P.DataLoss
A.SystemTime	Audit.3
A.AuditTrailProtection	Audit.4
A.AuthenticityServer	–
A.ArchivingSecrecyOfVoting	–
A.ProtectedCommunication	–
A.Buffer	T.DeletRecord

(E) Summary Security Problem Definition

The result from the comparison creates a discussion of uncovered threats and OSPs from Chap. 6, of vulnerabilities in the addressed trust model, and of the error handling and error recovery functionality in the PP.

Uncovered Threats and OSPs. An analysis of Fig. 8.1 – 8.3 reveals that some threats and organisational security policies from Chap. 6 are still uncovered in the GI/BSI/DFKI Protection Profile, including the assumption definition. For each uncovered threat and each uncovered OSP, a discussion of recommendations to include it or nor in a Protection Profile describing basic requirements and why or reasons to exclude it in such a core Protection Profile follows:

- *T.IntegVotes* - this threat is not needed in a core Protection Profile since the target of evaluation defines a system design where the tallying software is installed on the voting server and is started after the completion of the polling phase. In this case, a protection of the election data and particularly the votes is not necessary. Therefore, **removing this threat is acceptable**. However, see the recommendation on page 165 which discusses this design decision.
- *T.PersonalDataNet, T.SecretAuthNet* - Both of these threats are related. Both are caused by outside intruders who are sniffing the network. In the first case, the intruder wants to collect personal data by sniffing the

identification data that is transferred over the network, and in the second case, the intruder wants to obtain both identification and authentication data in order to use these to cast a vote on behalf of the voter. Both threats **should be added** because a corresponding security objective (O.GeheimNachricht) exists[21].

- *T.WrongSW* - The vulnerability of an outside intruder disseminating manipulated client-side voting software is not addressed in the security problem definition. However, in the security assurance section, the component ALC_ DEL.1 (Delivery procedure) is included, which demands that the delivery procedures are described and checked by the evaluator. Thus, the described vulnerability is reduced, which might be enough for a Protection Profile describing basic requirements. Therefore, **removing this threat is acceptable.**
- *P.Feedback* - This organisational security policy requires that feedback must be given to the poll workers in the case of exceptions, malfunctions, and breakdowns. The first two cases (that is, exceptions and malfunctions) can be implemented on the voting server, whereas the last case (that is, breakdowns) can only be ensured by an external component that checks the availability and the proper operation of the voting server. Because the GI/BSI/DFKI Protection Profile defines basic requirements, the method for handling breakdowns **does not necessarily need to be addressed**, while the other two aspects **should be added**[22] to the PP.
- *P.NoInteract* - This organisational security policy bars voter interactions in case of exceptions and malfunctions. This OSP **should be added** in order to ensure that votes do not get lost if problems arise.
- *P.SecrecyAfterBreakd* - In the event of any kind of exceptions, malfunctions, and breakdowns, this organisational security policy requires that the secrecy of the vote must also be ensured in such a way that the last elector cannot be linked to his vote and voters in the voting process cannot be linked to their selections. Breaking the secrecy of the vote, even for single voters, is a huge problem. However, organisational measures might be enough for a core Protection Profile. Therefore, a **corresponding assumption should be added** in order to ensure the overall objective of a secret election.
- *P.ClosePW* - This organisational security policy handles the end of the polling phase. Similar to elections in polling stations where the voter can arrive shortly before the polling station is closed and then cast his vote after the official closing time, this policy demands that the login to the voting server should be closed by the closing time, but those voters who are still in the voting process should have adequate time after the official

[21] This will be probably corrected in the final version as this inconsistency should be detected during the evaluation of the GI/BSI/DFKI Protection Profile.

[22] However, although this is not addressed in the PP's security definition, it is mentioned in the introduction.

closing time to cast their vote. This is an important aspect for elections at higher levels, especially in the case where remote electronic voting is applied in parallel to polling station voting and distance voting. However, since the GI/BSI/DFKI Protection Profile is intended to provide only a basis, **removing this OSP is acceptable.**

- *P.Spoil, P.SpoilWarning -* Both organisational security policies are labelled as non-core. Therefore, **removing this OSP is acceptable**.
- *Functional Requirements: P.StoreAllVotes, P.AccurRep, P.AccurDis, P.EqualRep, P.InfoPW, P.Transmission, P.AdequNoVotes, P.AdequatNoBallotOpt, P.CompatClient -*
  Some more organisational security policies are not treated in the GI/BSI/DFKI Protection Profile. These are pooled here because they all mainly address 'correct' functionality rather than security problems. Since the Common Criteria primarily address security aspects, a general discussion of whether to include or to exclude those requirements in a Protection Profile should be included. As the outcome of the discussion is open, for the moment, **removing these OSPs is acceptable.**

Insufficient Discussion of the Trust Model. The trust model is in some ways not explicit enough. Specifically, this concerns the threats T.ProofGen and T.SecretMessage. In these cases, the discussions about the consequences for the remote electronic voting system design from Sect. 7.2 should be considered. A decision for one of the provided cases to handle T.Proof and one of the cases to handle T.SecretMessage is still missing. This is an **essential improvement** because the current formulation allows the application of any of the provided cases. Thus, it is dependent on the evaluator's interpretation as to which cases are chosen. However, this would contradict the idea of comparable and repeatable interpretations.

Moreover, the functionality of the poll worker interface is not sufficiently discussed. Clarification is needed as to whether the poll workers must be present to access the TOE or whether a remote access is possible. If remote access is indeed possible, the text is ambiguous as to whether poll workers use the same remote access or whether they use different access channels. Moreover, if remote access is allowed, additional threats or assumptions need to be discussed and added (for instance, those threats and assumptions related to the device used for the remote access).

Insufficient Discussion of the Error Handling and Error Recovery. In the past, researchers concentrated on running remote electronic voting system that might be attacked. However, the cases where functional errors appear are not really discussed. Clarification is needed as to when a re-run should be possible and how to inform poll workers about problems. In this case, research is necessary, the results of which should be included in a future version of the core Protection Profile.

8.3.4 Security Objectives and Functional Requirements

Security Objectives and Security Rationale. Based on the recommendations of Sect. 8.2.3 a couple of changes and improvements result for the list of security objectives, both for the security objectives for the TOE and the security objectives for the environment, are suggested. Obviously, after having improved and extended the security problem definition and the list of security objectives, the security rationale section needs to be adjusted accordingly.

Security Functional Requirements and Security Functional Rationale. The next step concerns the PP section 'Security Functional Requirements'. According to the modifications in the list of security objectives for the TOE, this list must be checked for conformance and probably needs to be adjusted as well. Finally, the security functional rationale must be aligned to the improvements in the security objective section and the security functional requirements section.

8.3.5 Security Assurance Requirements

The analysis from Sect. 7.3 shows that the security assurance requirements identified in Sect. 6.5 map to EAL4+.

Open Points and Suggestions for Improvements. The discussion for the evaluation assurance level was mainly economically driven. Unfortunately, on the other hand, requiring a high EAL does not makes sense if no developer is going to use a corresponding Protection Profile. Moreover, the main discussion focused on the levels and less on the content; i.e., the discussion centred on what security assurance requirements are desired for elections with a basic Protection Profile. Thus, a recommendation is proposed to go through the whole catalogue with legal counsel and decide whether each component is needed and if so, in which concrete requirement does it belong. Having finished this process, the corresponding EAL level can be reconsidered.

8.4 Summary

This chapter focuses on the challenges in developing a Protection Profile that defines core requirements for remote electronic voting systems. Therefore, the GI/BSI/DFKI Protection Profile is described. Based on the findings from previous chapters, a few necessary improvements are identified.

The first two sections describe and discuss the GI/BSI/DFKI Protection Profile. Section 8.1 presents the background and history of the GI/BSI/DFKI project, including the involved parties and people. Then, this section illustrates the motivation to define 'only' core requirements. A common basis for any kind of election (that can be extended) should be provided instead of ultimately creating several Protection Profiles for different types of election that are not comparable. Afterwards, the main points of discussion in the project are showcased. These include the PP's title, the evaluation assurance level, the decision regarding whether to demand demonstrable or strict conformance, and the extension on the first page proposed by the Federal Ministry of the Interior.

The GI/BSI/DFKI Protection Profile itself is discussed in Sect. 8.2. The main parts of the PP introduction, the TOE overview, and the conformance claims (in particular strict conformance) are pointed out. Then, the content of the security problem definition is summarised and the list of security objectives addressed. In addition, the list of specified security functional and security assurance requirements (EAL 2+) is presented. For the two main issues, the conformance claim and the evaluation assurance level, the background for the decision is clarified.

A comparison between the GI/BSI/DFKI Protection Profile and the approach in this book is discussed in Sect. 8.3. Here in particular, the introduction section and the security problem definition part are addressed. Both approaches are first compared and then open points and hints for improvements are identified.

The GI/BSI/DFKI Protection Profile with the identified improvements constitutes the proposed evaluation methodology for remote electronic voting systems which can now be applied to available systems. It is the first evaluation methodology for remote electronic voting systems which is standardised (and, thus, produces comparable results), takes the underlying trust model into account, and is flexible with respect to different evaluation depths.

Part IV

Application

9

Proof of Concept

The previous part discusses the GI/BSI/DFKI Protection Profile which constitutes after the implementation of the identified improvements as the proposed evaluation methodology for remote electronic voting systems. The result can now be applied to available systems. Currently, there is no system that has been evaluated against the GI/BSI/DFKI Protection Profile or even against the improved version.

This chapter aims to gain experiences with the application of this evaluation framework. Thus, the Estonian system and the POLYAS system[1] are analysed with respect to this framework. Due to space and time constrains, no complete Common Criteria evaluation has been undertaken. It has been decided to evaluate against the security problem definition retrieved from the extended and improved core Protection Profile as described in the previous chapter. This analysis is based on a system description deduced from available documents. The result is provided in this chapter.

9.1 Procedure Specification

Due to time and space constraints, no formal Common Criteria evaluation is presented for the Estonian system and the POLYAS system. The provided analysis is based on the security objectives from the GI/BSI/DFKI Protection Profile and the recommended extensions from Sect. 8.3, while, with respect to the secrecy of the vote, it is assumed that it is sufficient to ensure the secrecy of the vote till the next election (see Sect. 7.2.2 for further discussions). In addition, the analysis considers the PP assumptions about the environment

[1] A similar analysis has been done in joint work with Hugo Jonker for the Dutch Rijnland Internet Election System (RIES) in [76]. However, RIES does not fit to the considered target of evaluation from Sect. 6.2 because it provides voter verifiability and ensure the secrecy of the vote in the election setup phase. Thus, it is not further discussed here.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 177–193, 2009.

and the intruder's technical capabilities as considered in Sect. 8.2.3. For each security objective, it is outlined whether (and if yes, with which TOE security function(s)) each of the two systems meets this particular security objective (in an adequate and sufficient way).

This analysis mainly corresponds to a Security Target evaluation (which is part of a Common Criteria evaluation). However, this analysis is based on the security objective, while a formal CC evaluation of the Security Target would be based on the security functional requirements. As Sect. 8.2.2 recommends using demonstrable conformance[2], this "easier" case is applied for the analysis.

Besides the main results (PASS, FAIL, and INCONCL[3]), the result ORG is applied to indicate that the developers are aware of corresponding problems or attacks but implement only organisational solutions to meet the corresponding security objective.

Requirements, which are extended, clarified, or added by the hints for improvements in Sect. 8.3, are labelled with an asterisk '*'. The detailed description of the processes and the protocols is separated from those during the election setup phase, those during the polling phase, and those during the result calculation process.

9.2 The Estonian System

Already in 2001, the Ministry of Justice announced intentions to introduce remote electronic voting. In 2005, remote electronic voting was implemented as an additional voting channel for local elections. Two years later, remote electronic voting for the Riigikogu (Estonian parliament) election was the first countrywide use of the Internet as a voting channel in a parliamentary election. There was no special registration process but each voter was able to vote using remote electronic voting. Even though there was no sign that the voters rejected remote electronic voting in the 2007 elections, only 5.4 percent of voters cast votes using the Internet as their voting channel.

According to the legislation, remote electronic voting is allowed under three main preconditions: firstly, the voter has to identify and authenticate himself with his digitally-enabled ID card[4], secondly, remote electronic voting is implemented as advance voting (from six to four days before election day), and thirdly, vote updating is enabled (in particular after having cast an electronic vote, the voter can overwrite this vote by casting a paper vote in an

[2] The necessary explanations as demanded for demonstrable conformance are left out.

[3] INCONCL means that the available sources do not provide enough information to determine any of the other verdicts.

[4] In Estonia, the new and already broadly distributed personal identification document (ID card) contains a chip which enables the user to be identified via the Internet and to digitally sign legally accepted documents.

advanced polling station). All technical activities related to the remote electronic voting process were audited by an external auditing company KMPG Baltics, including the election setup phase, polling phase, and tallying phase. The audit was performed against written documents describing the necessary steps and procedures.

The following system description and analysis are based on the following documents:

- OSCE/ODIHR Election Assessment Mission Report for the Parliamentary Elections of Estonia [106]
- The paper "Towards Remote E-Voting: Estonian case" [94]
- The paper "E-Voting in Estonia 2005. The First Practice of Country-wide Binding Internet Voting in the World" [95]

9.2.1 System Description

a) Classification

According to the classification from Sect. 2.1 the Estonian system can be classified in the following way:

- The Estonian System belongs to the *remote electronic voting system* category.
- The identification and authentication technique in use is a combination of *possession-based* and *secret-based*; in particular, the Estonian ID card is used, which identifies the voter over the Internet and enables the voter to digitally sign documents (for instance, his encrypted vote). To use this functionality, the voter needs to know his two PIN codes associated with the ID card: one for identification and one to sign documents.
- With respect to the secrecy of the vote the Estonian system is a representative of the class anonymity is ensured in the tallying phase by applying a hardware security module.
- The Estonian System belongs with respect to the different client-side voting software classes to the *fat-client* approach: there are three types of client-side voting software for the three different operating systems namely Windows, UNIX, and Apple MacOS.

The Estonian system does not match exactly the TOE description from Sect. 6.2 as it enables vote updating and allows the responsible election authority to change the electoral register (and in fact they did this every day). For the following analysis, this additional functionality is not considered.

b) Overview

From an abstract level, the Estonian system works in the following way: the voter logs onto the voting server and identifies himself with his ID card (using PIN1). Then, the voting server checks the voter's identity and provides the corresponding ballot to the voter. After having made his choice, the voter digitally signs his encrypted vote. The voting server verifies the voter's signature. In the tallying phase, first, the digital signature is removed, then the encrypted votes are scrambled, and finally, they are decrypted by the HSM and counted. The main parties in the Estonian system are as follows:

- Registration server (RS)
- Certification server (CA)
- Vote storage server (VSS)
- Counting software on a separate PC (CPC)
- Hardware security module (HSM)
- Client-side voting software (CSS)

In the Estonian remote electronic voting system, the *one* voting server is further separated into three servers, namely the registration server, the vote storage server, and the certification server, while the last one is involved but not set up in particular for the remote electronic voting solution but for any application based on the digital identity card. The tallying software is partitioned in one part, running on the tallying PC and a second one running on the hardware security module to decrypt the votes. The Estonian System implements the following communication links:

- CSS - RS: to communicate with the remote electronic voting system
- RS - VSS: to forward votes to the storage
- VSS - CA: to check whether the voter's certificate is still valid.

c) Description of the Election Setup Phase

Preparation on the Server-Side. Preparation on the Server-side. On the server-side, several steps must be taken as follows: new RS, VSS, and CPC are purchased and reinstalled with an operating system, security mechanisms (for instance firewalls), and the corresponding voting/tallying software. The Hardware Security Module is set up; that is, a key pair is generated. While the secret one is stored on the device, the public key is integrated in the client-side voting software. Moreover, keys to enable the HSM are generated: seven keys, which are distributed to the National Election Commission (NEC) members, and two for the administrators. These keys are generated in a way that the two administration keys and four out of the seven NEC keys are necessary to enable the HSM.

Preparation on the Voter-Side. Preparation on the Voter-side. The voter needs to be prepared to use the electronic channel. Besides his electronic identity card, he needs to have a corresponding smart card reader and needs to know his two PINs. Moreover, if he uses MacOS or Linux, the voter needs to download the client-side voting software. In the case of a windows user, he needs to have Java enabled, so that the web browser can load the corresponding Java Applet.

d) Description of the Polling Phase

The high-level protocol steps during the polling phase are described in Fig. 9.1. This figure uses many shortcuts, therefore some explanations are given here:

- *SSL* – Two directed SSL connection (with the voter's first secret key enabled with PIN1).
- *elig*? - Here the RS checks whether the requesting person is an eligible voter.
- $re-vote$? - Here the VVS checks whether the requesting voter has already cast a vote[5].
- *gen ballot* - generate ballot that belongs to this particular voter.
- *choose* - the voter makes his choice.
- *vote* - the system displays the voter's choice and the voter verifies whether he wants to confirm this choice or changes his choice again.
- $sig(m)$ stands for signing the message m with the voter's secret key enabled with PIN2 while such a message is implicitly extended with the voter's certificate for the corresponding secret key.
- $enc(m)$ stands for encrypting a message m with the public key from the HSM.
- $sig-ID$ - the RS verifies whether the signature belongs to the person that started the session.
- *sig ok* - the VVS verifies the signature and the validity of the certificate.

e) Description of the Tallying Phase

After the electronic polling phase and closing the advanced polling stations, those e-votes stored at the VS where voters also cast a paper vote, are labelled with"not to be counted". Then a CD is burned containing the last received e-vote per voter in a randomised order (while those labelled with "not to be counted" are excluded). This CD is sealed and handed over to the NEC chairman. On election day, one hour before the polling stations close, the result calculation process starts. The e-votes are loaded on the CPC, which is connected to the HSM (via cable). Next, the HSM is enabled by entering four of the seven NEC keys and the two administration keys. Now the encrypted votes are sent to the HSM vote by vote, and the HSM sends corresponding decrypted votes back. Having finished the decryption, the votes are tallied and the result is burned onto another CD. This CD is loaded onto an other ordinary PC in order to display the result in a human readable way. The result is digitally signed by the NEC chairmen. The signed result is the legal one.

[5] In case, the voter has already cast an e-vote, this information is displayed to him and he is asked whether he wants to update his vote.

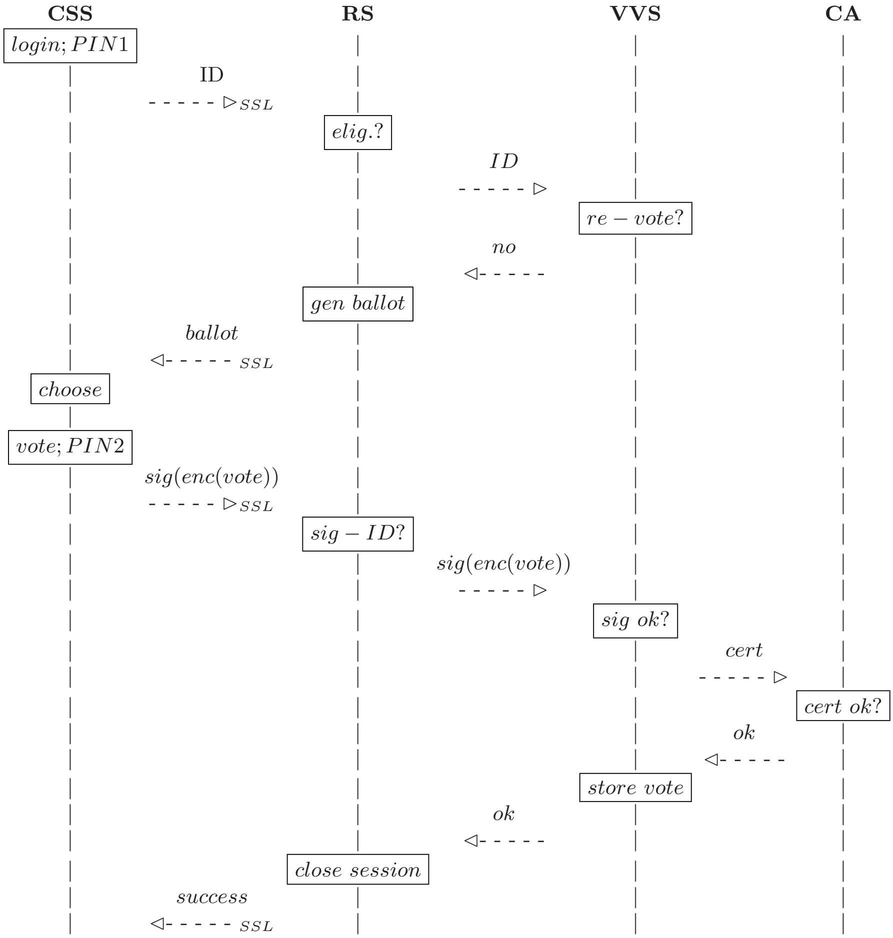


Fig. 9.1. The voting protocol implemented in the Estonian system

9.2.2 System Analysis

Based on this information, the identified security objectives are checked to see if they meet the requirements from chapter 8. The result of this evaluation is summarised in Table 9.1 (for O.T.) and 9.2 (for O.OSP.).

Result. The tables show that the Estonian system meets most of the security objectives with a PASS (at all 17). Two are met by organisational means and for seven of the security objectives no statement is possible due to missing information about the system. The inconclusive security objective only affect those objectives deduced from organisational security policies. As there is also no FAIL in the result, there is currently no reason that a formal Common Criteria evaluation of The Estonian System against the BSI/GI/DFKI Protection

Table 9.1. Result of the analysis for the Estonian system (part 2)

Security Objective	Result	Explanation
O.T.IneligVoter	PASS	The identification and authentication is based on the voter's digitally-enabled ID card.
O.T.OneVoterOneVote	PASS	The Estonian system implements vote updating, thus, Estonians are allowed to cast more than one vote. However, the system ensures that only the last vote is taken for the tallying. Note, this security objective would FAIL in the case of a strict conformance claim.
O.T.ProofGen	PASS	According to Fig. 9.1, it is not possible to generate a proof from any information either sent to, displayed on, and/or sent from a vote-casting device.
O.T.DeleteMsgNet	PASS	According to Fig. 9.1, this is ensured as long as the voter ensures that he receives the last confirmation.
O.T.AlterMsgNet	PASS	The communication is secured by SSL. In addition, votes are signed by the voter.
O.T.ElectionSecrecyNet	PASS	The communication is protected with SSL. In addition, votes are encrypted with the public key of the HSM.
O.T.IntResultNet	PASS	See O.T.ElectionSecrecyNet
O.T.WrongServer	PASS	As the Estonian system uses SSL, this security objective is ensured as long as the voter verifies the server certificate.
O.T.IntegElecData	ORG	After closing the poll, a CD is burned containing the last received e-vote per voter. This CD is sealed. Thus, the integrity of e-votes is only ensured by organisational means. In addition, the protected data only contains e-votes, while it is required to protect any kind of election data.
O.T.ElectionSecrecy	PASS	The encrypted e-votes are stored on the CD in a randomised order and without the voter's signature (anonymousness by scrambling the e-votes). After the tallying phase, there exists a second CD containing the list of decrypted votes. However, as it is not stored, it is unknown which encrypted e-vote from the first CD belongs to which voter, so O.ElectionSecrecy is ensured.
O.T.PersonalDataNet*	PASS	The identification data (ID) sent in the first steps of the protocol is secured with SSL.
O.T.SecretAuthNet*	PASS	It is not necessary to protect the authentication information on the Internet as only the voter can sign votes with his private key.

Table 9.2. Result of the analysis for the Estonian system (part 1)

Security Objective	Result	Explanation
O.OSP.Interface	PASS	All required functionality is implemented.
O.OSP.PWClosePoll	INCONCL	–
O.OSP.PWInterface	INCONCL	–
O.OSP.Confirmation	PASS	See Fig. 9.1.
O.OSP.SelfCheck	INCONCL	–
O.OSP.ErrorRecovery	INCONCL	–
O.OSP.Auditing	INCONCL	The Estonian system produces audit data, but it is not known which information is stored.
O.OSP.VoteRight	PASS	This is ensured by the implementation of vote updating.
O.OSP.VoteRightExc	INCONCL	–
O.OSP.SepDuty	PASS	Two administrators need to enter their passwords.
O.OSP.AC	ORG	There was an AC mechanism implemented on the voting server.
O.OSP.AccurCalc	PASS	This was shown by tests in advance of the election.
O.OSP.Feedback*	PASS	The administrators are informed via SMS.
O.OSP.NoInteract*	INCONCL	–

Profile should fail. In particular, there is no reason to change the architecture or the voting protocol in order to get the system certified. Minor changes with respect to the INCONCL security objectives might be necessary.

9.3 The POLYAS System

The POLYAS system is the voting system from a company called Micromata GmbH. It has a long-standing history – compared to the field itself – which starts in 1996, where the first election was carried out with 64.000 young Finnish pupils. Nowadays, the POLYAS system has been used to cast more than 340.000 votes, 210.000 of which were in Germany. In the last years, the system has been improved continuously by a close partnership with the Gesellschaft für Informatik (GI - the German society of computer scientists) and here the advisory board of security and voting experts. The GI has used the POLYAS system in parallel to postal voting for their yearly held elections since 2005. Beside several GI elections, the POLYAS system was also used for the elections of the Deutsche Forschungsgemeinschaft (DFG - German Research Foundation) in 2007.

For the system description and the analysis the following documents have been used:

- The paper [123][6] at the Vote-ID conference
- The POLYAS system Web page (www.polyas.de)
- Two confidential manufacturer's documents: one document describes how the POLYAS system ensures the requirements from the GI requirement catalogue [113], and the other one describes the procedure to activate the POLYAS system

The POLYAS system is described and analysed in the same software version as used for the GI elections.

9.3.1 System Description

a) Classification

According to the classification given in Sect. 2.1 the POLYAS system can be classified in the following way:

- The POLYAS system belongs to the *remote electronic voting systems* according to the defined election forms.
- The identification and authentication technique in use is *secret-based*[7]; in particular, the GI membership number is used to identify the voter and the authentication token is generated in the election setup phase and sent to the voter via ordinary mail.
- With respect to the secrecy of the vote, the POLYAS system is representative of the class *anonymity is ensured during the polling phase* and of the sub-class *separation of duty principle.*
- With respect to the different client-side voting software classes, the POLYAS system belongs to the *Web browser solution* approach; it supports any Web browser, including "lynx" a text-based Web browser.

b) Overview

The main parties in the POLYAS system are as follows:

- The client-side voting software (CSS)
- The electoral register server (ERS)
- The validator server (VS)
- The ballot box server (BBS)

In the POLYAS system, the *one* voting server is separated into three different servers, while each server is located at a different place and administrated by a different party.

[6] This paper is based on a contracted study developed by the e-voting.cc competence center and the author of this book.

[7] In the POLYAS implementation for the D21 elections, the identification and authentication technique is based on digital signature cards.

The POLYAS system implements the following communication links:

- CSS - ERS: to check the voting right,
- VS - ERS: to control the ERS's decision about the voter's voting right and to generate random anonymous authorisation tokens T,
- CSS - BBS: to cast a vote, and
- ERS - BBS: to inform the BBS about valid authorisation tokens and vise versa to inform the ERS about unauthorised tokens because corresponding votes have been cast.

c) Description of the Election Setup Phase

The election setup phase contains the following six main tasks:

1. Generation of the authentication token (TAN):
   The process[8] generating the TANs has as output only the hashed ($hash(TAN_i)$) and encrypted ($encr(sk_P, TAN_i)$) TANs.
   - ($hash(TAN_i)$) is linked to voter$_i$ in the electoral register (containing all membership numbers to identify the voter). This register is stored on the ERS.
   - ($encr(pk_P, TAN_i)$) is linked to voter $_i$ in another copy of the electoral register (containing the voter's addresses but not their membership numbers).
     This extended electoral register is sent to the provider. In order to prepare the election material for the voters, the provider decrypts the TANs with his secret key sk_P and prints the TANs on the election material. This material is sent to the voter.
2. The following three key pairs are generated per server:
   - Https key pair (only for ERS and BBS)
   - Communication key pair
   - Database key pair

   Each of the https secret keys is stored on the corresponding servers as well as corresponding public https keys from the other servers needed to later verify messages. The https public keys from the ERS and the BBS are made public to the voter (on the Web page and printed in the election material). The voter can use these two keys to later verify whether he communicates with the proper servers.
   The public and private communication and database keys are stored on the corresponding servers. The corresponding secret keys are encrypted with two pass phrases, in a way that both are necessary to decrypt the keys. The six pass-phrases are each known by one of the six different members of the responsible election authority.
3. In the electoral register (containing the membership numbers - ID), corresponding authentication tokens (TANs) are added in the following hashed and signed way:

[8] This process is not further discussed with respect to its security functions.

$$ID - -hash(TAN) - -sig_{ERS} - -sig_{VS}$$
where $sig_{ERS} := sig(sk_{ERS}, hash(TAN))$ and
$sig_{VS} := sig(sk_{VS}, sig_{ERS})$;
the secret keys are those from the communication key pair.

This electronic electoral register is stored on the ERS. It is also hashed, signed with sk_{ERS}, and then stored in a secure way outside the system.

4. The ballot is designed, and the rules to cast a valid vote are defined. Both sets of information are stored on the BBS.
5. Two access tokens are generated for each of the servers to get general access to the servers in order to, for instance, store the electoral register or start the polling phase. Again, here are six different secrets, which need to be distributed amongst the responsible election authority.
6. The servers are configured and secured (for instance, by installing a firewall and a virus scanner). Afterwards, the corresponding POLYAS Software for the particular server is installed.

The amount of keys and their distribution is shown in Fig. 9.2.

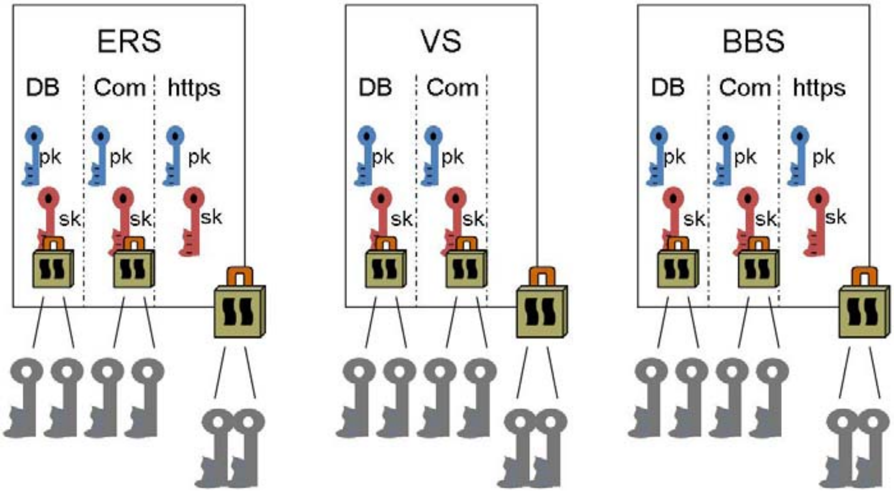


Fig. 9.2. Key distribution for the three POLYAS server

d) Description of the Polling Phase

First, the corresponding members of the responsible election authority log onto the BBS, using the authentication tokens in order to start the software. To do so, the other members responsible for this server need to enter their pass phrases to decrypt the database and the communication secret key (see Fig. 9.2). The same procedure needs to be taken for the VS. When both servers and the software run successfully, the pair knowing the access tokens for the

ERS uses these pass codes to log on and to start this software. Again, to do so, the other members involved need to enter their pass phrases to decrypt the database and the communication secret key. At the official beginning of the election, they start the polling phase using corresponding functionality of the ERS based on the POLYAS software.

The high-level protocol steps during the polling phase are described in Fig. 9.3. This figure uses many shortcuts, therefore some explanations are given here:

- SSL_i highlights all SSL communications of one session. In order to successfully cast a vote, four sessions are necessary (between different voting server components).
- *elig*? - Here the ERS checks whether the TAN corresponds to the ID and whether the corresponding voter has not cast a vote, yet.
- *checksig* verifies both received signatures (signed messages).
- *gen.* - means generate.
- T stands for the generated random authorisation token, which enables the voter to communicate anonymous with the BBS.
- *setinval.* - with this function the VS labels the value sig_{ERS} as invalid, that is, if there will be a second request from the ERS for a particular voter, the VS cancels the protocol (and in particular does not generate a new TAN T).
- *choose* - the voter makes his choice.
- *accept* - the voter confirms his choice (for the first time).
- *vote* - the system displays the voter's choice and the voter verifies whether he wants to confirm this choice or changes the choice again.
- $vote := choice$ - In step *store* the voter's choice has already been stored in a database. In this step this choice is labelled as vote. At the end, only labelled database entries are tallied.

In addition, the communication between the servers is secured by SSL using the corresponding communication keys. All votes and voting tokens are stored in an encrypted and signed manner, using the public key of the involved database. Moreover, votes are stored in a randomised order in blocks of 30. As soon as one block is completed, the corresponding votes are concatenated to one string, which is hashed and published. The next block will be treated similarly but built as a hash-chain:

$$hash(permut(vote_1, .., vote_{30}))$$
$$hash(hash(permut(vote_{31}, .., vote_{60}))\#hash(permut(vote_1, .., vote_{30})))$$
$$...$$

e) Description of the Tallying Phase

To close the election, again the members of the responsible election authority who have the authentication tokens need to log onto the corresponding servers (see Fig. 9.2). In the next step, first the ERS is taken off-line followed by

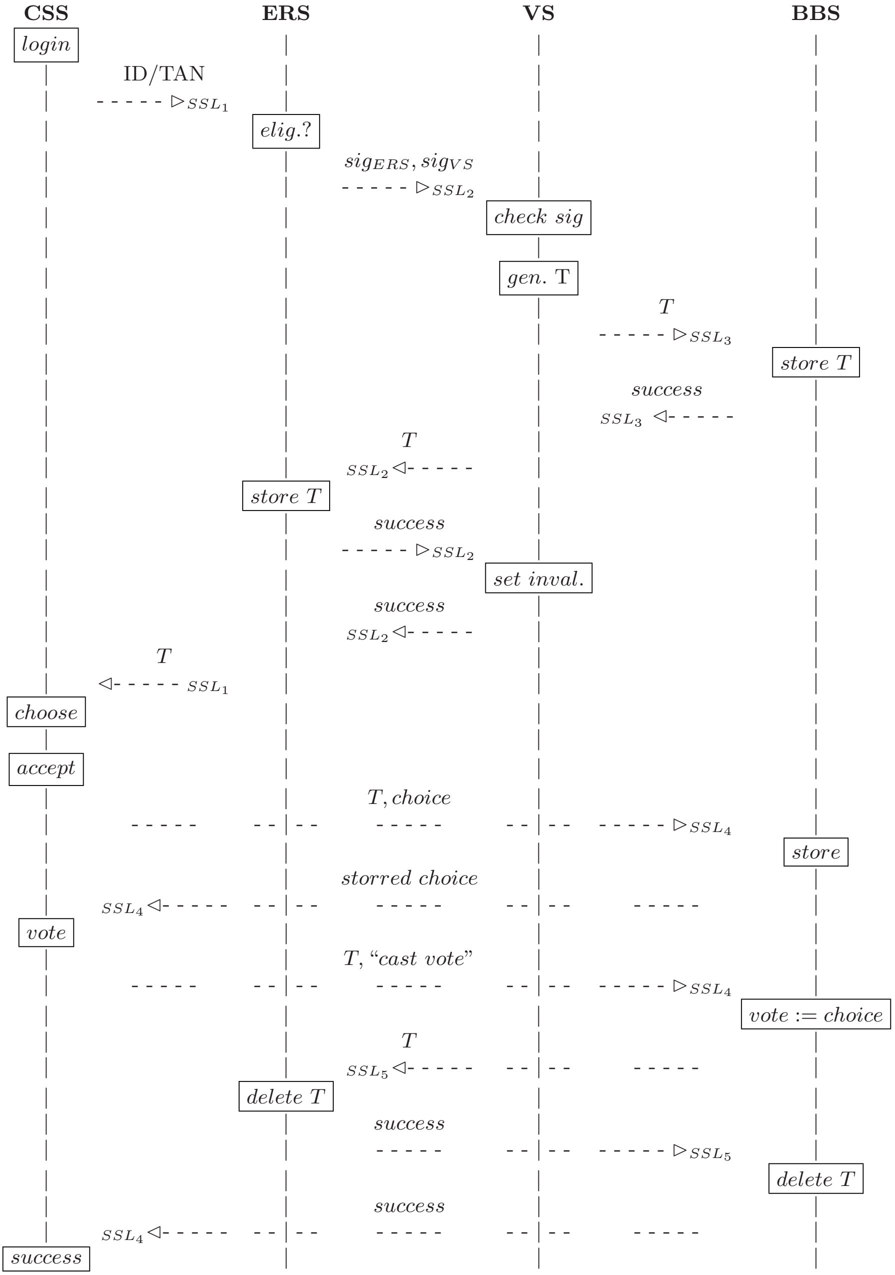


Fig. 9.3. The POLYAS voting protocol

the other two servers. Afterwards, the member of the responsible election authority needs to enter the pass phrase for the secret database key of the BBS. Thus, the e-votes can be decrypted, and the tallying software can calculate the election result.

9.3.2 System Analysis

A related analysis has been done by the developer in [123]. Here, the POLYAS system is analysed according to an older version of the GI/BSI/DFKI Protection Profile. The adoption of the analysis result from [123] and version 0.27 of the GI/BSI/DFKI Protection Profile is presented in Table 9.3 and 9.4. Several times, results from [123] are cited (together with a label from the corresponding security objective in this paper).

Result. There are some problems in deciding upon PASS or FAIL because the POLYAS system only provides a responsible election authority interface to start the election, while to stop the polling phase and to start the tallying shell, commands are used. The quantitative result shows that almost all security objectives deduced from threats got a PASS (10 of 12), while there is only one FAIL and one ORG. With respect to the security objectives related to organisational security policies, 7 security objectives are evaluated to PASS, 4 to ORG, 1 to FAIL, and 2 to INCONCL. The security objectives that are evaluated to FAIL are O.T.ElecSecrecyNet and O.OSP.Auditing. For all identified problems, [123] claims to have a solution. Thus, minor changes are necessary, in order to certify the POLYAS system, while these modifications are not related to the architecture or the voting protocol steps.

9.4 Summary

This chapter applies the developed evaluation framework (which is based on improvements to the GI/BSI/DFKI framework) to the two available systems: the Estonian system and the POLYAS system. As the analysis is different from a complete Common Criteria evaluation, Sect. 9.1 describes the applied evaluation procedure. This procedure is mainly based on the security problem definition. Moreover, the analysis for both systems starts with a detailed system description.

Section 9.2 addresses the Estonian system and Sect. 9.3 examines the POLYAS system. Based on the security problem definition, both systems (at least in the analysed version) do not meet all the specified security objectives. However, only minor modifications are necessary for both systems to be compliant[9], while the improvements are not related to the voting protocol.

[9] This compliance makes only statements on the system providing enough security functions to meet the security objectives (which corresponds to a Security Target evaluation), while a Common Criteria evaluation according to EAL2+ contains much more security assurance requirements.

Table 9.3. Result of the analysis for the POLYAS system (part 1)

Security Objective	Result	Explanation
O.T.IneligVoter	PASS	”It is only possible to get a voting token enabling a voter to cast a vote after sending the ID and the voting TAN to the ERS, which decides whether the request comes from an eligible voter who has not yet cast a vote. Without having such a valid voting token you can send vote messages to the BBS but these are rejected.” (SecObj1)
O.T.OneVoterOneVote	PASS	See O.T.VoteRight (SecObj26)
O.T.ProofGen	PASS	According to Fig. 9.3, it is not possible to generate a proof from any information either sent to, displayed on, and/or sent from a vote-casting device. (SecObj2)
O.T.DeleteMsgNet	PASS	According to Fig. 9.3, this is ensured by SSL as long as the voter verifies the server certificate and ensures that he receives the last confirmation. (SecObj3)
O.T.AlterMsgNet	PASS	See O.T.DeleteMsgNet. (SecObj3)
O.T.ElectionSecrecyNet	FAIL	”First of all, the vote is transmitted encrypted via SSL. Secondly, the vote is not sent together with the identification data, not even during one SSL session. Thus, one can only link the encrypted identification data to the encrypted vote via corresponding sender IP addresses. The current problem is that someone who is observing the Internet and knows, which IP-address a particular voter has, can limit the possible choices the voter makes because of the size of the vote message. Especially, casting an invalid vote by choosing all candidates is observable.” (SecObj4)
O.T.IntResultNet	PASS	See O.T.DeleteMsgNet. (SecObj5)
O.T.WrongServer	PASS	As SSL is used, this security objective is ensured as long as the voter verifies the voting server’s SSL certificate.
O.T.IntegElecData	ORG	”After the completion of the result computation, POLYAS computes a hash value of the electoral register (including those who cast a vote and who did not) and a hash value of all votes. These two hash values are printed immediately and are part of the election commission documentation, which is signed by the election commission.” (SecObj6)
O.T.ElectionSecrecy	PASS	”The only link between a voter and his vote on the server-side is the voting token. But the voting token is deleted at the ERS and the BBS just after completing the voting process for the corresponding voter. Thus, even knowing all data from the servers after the election it is not possible to compromise the secrecy of the vote because the link was already removed during the election.” (SecObj7)
O.T.PersonalDataNet*	PASS	The identification data (ID) sent in the first steps of the protocol is secured with SSL. (SecObj5)
O.T.SecretAuthNet*	PASS	See O.T.DeleteMsgNet.

Table 9.4. Result of the analysis for the POLYAS system (part 2)

Security Objective	Result	Explanation
O.OSP.Interface	PASS	All required functionality is implemented. (SecObj8/19/20/22)
O.OSP.PWClosePoll	ORG	"At the particular day and time the election commission meets in order to first deactivate the VS and the BBS and later the ERS. But it is not controlled by POLYAS whether the end of the election is already reached." (SecObj9)
O.OSP.PWInterface	PASS ORG	"there is no functionality implemented for the election commission to access the (encrypted) votes [...]" (SecObj11), "[...] to access the database containing the (encrypted) votes (other than for the result computation) [...]" (SecObj12), and "[...] to access the electoral register [...]" (SecObj14/15/17). However, the reset functionality (SecObj13/16) and, thus, the calculation of intermediate results (SecObj18/30) is only ensured by organisational means.
O.OSP.Confirmation	PASS	See Fig. 9.3. (SecObj21)
O.OSP.SelfCheck	ORG	"Before the election each part of the software is digitally signed, meaning at any time the two election commission members responsible for a particular server can access the server and check whether the software running is still the one that has been installed. Moreover, the servers are observed using the Nagios software. This software checks regularly whether the server and the databases are still online and available."(SecObj23)
O.OSP.ErrorRecovery	ORG	"A comprehensive and exhaustive recovery concept has been developed containing all possible breakdown and restart scenarios. In case of system breakdowns, including data loss the election commission is informed and possible actions are discussed (is a restart possible?)."(SecObj24); also (SecObj28)
O.OSP.Auditing	FAIL	"Most of the events listed above are logged by POLYAS. The election data stored at the beginning of the election and the results after the counting process are missing in the current version. The audit records can be read on the corresponding server." (SecObj25)
O.OSP.VoteRight	PASS	"The POLYAS software installed on the ERS ensures that only those voters having valid IDs and voting TAN can continue the voting process and then cast a vote. It also ensures that all such voters can continue the voting process." (SecObj26)

Table 9.4 (continued)

Security Objective	Result	Explanation
O.OSP.VoteRightExc	PASS	"[...] the one voter-one vote principle can be ensured for all these situations as long as the voter takes care that in the event of not having received the final receipt, he or she needs to re-login to complete the voting process." (SecObj27)
O.OSP.SepDuty	PASS	Two administrators need to enter their passwords.
O.OSP.AC	ORG	The voting server's access control is used. (SecObj29)
O.OSP.AccurCalc	PASS	"The source code has been examined by the Physikalisch-Teschnische Bundesanstalt (PTB). They especially checked the vote casting algorithm."(SecObj31)
O.OSP.Feedback*	INCONCL	–
O.OSP.NoInteract*	INCONCL	–

Making the required modifications would mean that both systems could get certified in general.

As both systems are based on different architectures, different authentication techniques, different approaches to ensure the secrecy of the vote, and different implementations for the client-side voting software, it can be concluded that the proposed evaluation framework is very flexible. Moreover, no improvements for the framework can be deduced from this analysis.

10

Separation of Duty Principle

In Chap. 9, two existing remote electronic voting systems are analysed (the POLYAS system and the Estonian system) according to the evaluation framework presented in Chap. 8. In terms of a proof of concept it is shown that the framework is suitable for remote electronic voting systems and flexible enough to cover arbitrary systems. In addition, according to Chap. 7, the Common Criteria Protection Profile overcomes the identified vulnerabilities from existing requirement and evaluation documents because

- it is based on a standardised, consistent, and exhaustive list of requirements,
- the Common Criteria is an internationally accepted evaluation standard (ISO 15408) that strictly guides the evaluator with the Common Evaluation Methodology, and
- the Common Criteria is flexible with respect to different trust models and different evaluation depths.

However, the Protection Profile only considers two aspects of the trust model, assumptions to the environment and the intruder's technical capability. Therefore, in this chapter, the third aspect of the trust model – who can be trusted not to maliciously cooperate with others – takes centre stage. It is shown that this aspect cannot be integrated in the Protection Profile without losing the flexibility to meet different implementations of remote electronic voting systems.

Thus, an independent evaluation methodology to measure the separation of duty level for remote electronic voting systems is presented: the computation of the k-resilience value. This approach is exemplarily applied to some aspects of the POLYAS system and the Estonian system.

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 195–202, 2009.

10.1 Motivation

In the core Protection Profile presented in Chap. 8, the third aspect of the trust model definition, namely who can be trusted not to maliciously cooperate with others, is only addressed for the poll worker interface in O.OSP.SepDuty. All other aspects such as on the voting server, administration, or development level are only indirectly covered by assumptions about the environment.

The following assumption supposes that a separation of duty on the administration level does not need to be addressed:

A.PollWorker: The poll workers are trustworthy and they only access user and TSF data using the server-side TOE that is, they only use the functionality provided by the TOE[1].

A.PollWorker assumes that no single poll worker who has physical access to the voting server will bypass the server-side voting software. This assumption is much stronger than the corresponding one for traditional elections where several poll workers are on duty at the same time in order to "control" each other. Traditionally, it is assumed that some poll workers do not maliciously cooperate with other poll workers but at least one in each group is honest. In A.PollWorker, it is assumed that all poll workers are trustworthy.

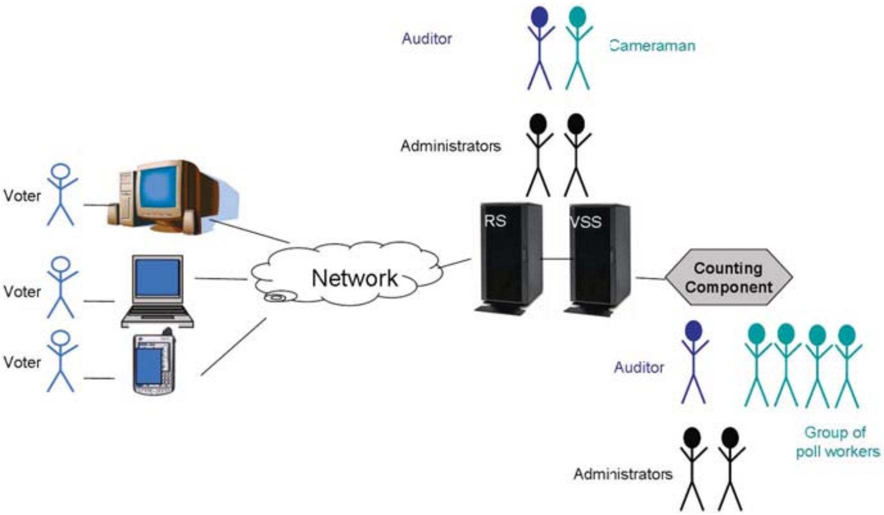


Fig. 10.1. Main system architecture of the Estonian system

[1] The assumption A.Wahlvorstand (=A.PollWorker) is translated from the German Protection Profile.

There are two main ways to ensure assumption A.PollWorker:

- Organisational measures, such as those used in traditional elections and in Estonia with the auditor, the cameraman, and the policeman (see Fig. 10.1). These ensure that only the two poll workers (here also called administrators) get access to the voting server and observe the poll workers' activities at the voting server.
- Implementing separation of duty in the system architecture as applied in the POLYAS system (see Fig. 10.2). Within this approach, the voting server is split into several voting servers; all hosted and administered by different groups of poll workers. Now, at least one poll worker per voting server needs to be malicious and bypass the server-side voting software. Thus, such a distributed architecture increases the number of malicious poll workers needed for a successful attack compared to a system implementing only one voting server.

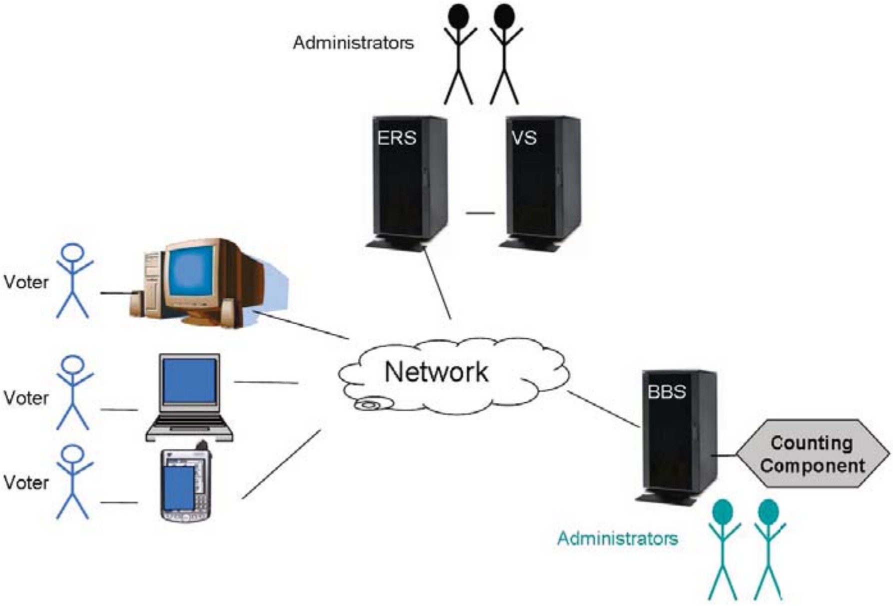


Fig. 10.2. Main system architecture of the POLYAS system

The first approach is easy to apply after the deployment of the system, while the second one needs to be considered in the design phase of the remote electronic voting system. The separation of duty approach is common and established in the research community as proposed voting protocols usually work with a generic number n of voting servers while k out of n needs to be manipulated by malicious poll workers in order to manipulate the system undetected. Note, more voting servers do not necessarily mean that more people need to act corruptly; the separation of duty principle must be implemented correctly.

The following PP assumption supposes that a separation of duty on the voting server level does not need to be addressed:

A.VotingServer: Protection of the voting server against attacks that originate from the insecure network is provided by the application of a security concept for the network connection, which prevents access to the voting server from network attackers.

Thus, while A.PollWorker covers voting server manipulations from an inside intruder (namely a poll worker), A.VotingServer covers voting server manipulations from an outside intruder attacking the voting server over the network.

There are two main approaches to ensure this assumption:

- Organisational measures by securing the one centralised voting server as much as possible.
- Implementing separation of duty in the system architecture: in the case that a remote electronic voting system works with n voting servers while k out of n need to be manipulated for a successful attack, the effort for an outside intruder who tries to get access to the voting servers increases: now, he needs to gain access to k out of n voting server(s). Therefore, it is essential not to use the same security concept on all voting servers as this would not increase but merely replicate the effort needed.

The implementation of separation of duty in the system architecture also has advantages with respect to malicious developers (as the third group of attackers beside inside and outside intruders): the different server-side voting applications can be implemented by different developers (even from different companies).

Result. This discussion shows that depending on the chosen realisation of the two addressed assumptions, the underlying trust model in terms of the separation of duty aspect of a particular remote electronic voting system can differ. However, this is not addressed in the core Protection Profile from Chap. 8. It is possible to certify remote electronic voting systems that do not implement the separation of duty principle on the system architecture level.

There are two reasons why the demand for the implementation of the separation of duty principle in the system architecture is not integrated into the core Protection Profile:

- The Common Criteria does not allow variables such as k out of n in a Protection Profile. Using a constant is not an option, as proposed voting protocols implement different numbers of voting servers. Thus, the only possible to keep the core Protection Profile flexible for any kind of voting protocol is to subsume different voting servers to one main voting server.

- Even if the realisation of the separation of duty principle in system architectures seems essential, in the end it is the responsible election authority that has to decide whether organisational measures such as those used in Estonia are sufficient. The idea of the Protection Profile is that all remote electronic voting systems need to be compliant at least to this core Protection Profile. Therefore, the separation of duty aspects should not be required here.

Nevertheless, for the responsible election authority, it is essential to know the number of malicious entities needed to successfully attack the remote electronic voting system. An entity can be a voting server, a poll worker (including persons performing any server configurations and administrators), or developers of the voting software.

The separation of this aspect and the CC certificate has two advantages: the importance of the separation of duty aspect of the trust model is highlighted and the independence of both aspects - separation of duty and the CC evaluation according to the PP - is underlined. The Common Criteria framework provides confidence that from the technical point of view the system enables secure elections (in environments where the assumptions and the intruder's technical capability hold), while the k-resilience value helps to undertake additional organisational measures to ensure corresponding assumptions about the environment hold.

10.2 'k-resilience' Approach

Driven by this discussion, it is recommended to extend the evaluation framework from Chap. 8 by the computation of the $\langle(k_{1,1}+ \ldots +k_{m,1}$ out of $n_{1,1}+ \ldots +n_{m,1}), \ldots, (k_{1,i}+ \ldots +k_{m,i}$ out of $n_{1,i}+ \ldots +n_{m,i})\rangle$-resilient value[2]. This value and its computation are explained stepwise (for a similar discussion see also [154]).

Step 1: A system is called k-resilient with respect to a particular security objective[3] if at least k entities need to be corrupted in order to successfully compromise this security objective. No subset of $k-1$ entities can violate the security objective. So, if a system relies on a single trusted entity with respect to a particular security objective it is just 1-resilient with respect to this security objective.

[2] This idea is based on discussions with Berry Schoenmakers about the vulnerabilities of the Common Criteria approach.

[3] The separation of duty aspect is not relevant for all security objectives but applies to the following security objectives from Sect. 6.3: O.T.UnauthVotes, O.T.TamperServer, O.T.ElectionSecrecy, O.T.AffectCounting, and O.T.AC.

Step 2: Note that k-resilience is relative to the total number of entities – 1-resilience with 10 involved entities is weaker than 1-resilience with 5 involved entities. Therefore, it is recommended to determine for all security objectives the total number of entities n and call it 'k out of n'-resilient with respect to a particular security objective.

Step 3: There is another aspect which needs to be taken into account as the set of involved entities is heterogeneous (meaning not all entities are equal) – for example, in the case of 2 out of 5-resilient systems, it might happen that for one of these two necessary malicious entity you can choose '1 out of x' and for the other one you can choose '1 out of y' (while x and y are different; in addition the corresponding sets can be disjoint). To display this in the specified value, the value is extended in the following way: '(k_1+...+k_m out of n_1 +...+n_m)' with respect to a particular security objective.

Step 4: For some security objectives there might be different possibilities in some systems to violate this security objective; for example, either by corrupt entities A and B or by corrupt entities C and D. To also cover this case, the last value is once more extended to '$\langle(k_{1,1}$ + ... + $k_{m,1}$ out of $n_{1,1}$ + ... + $n_{m,1})$, ..., $(k_{1,i}$ + ... + $k_{m,i}$ out of $n_{1,i}$ + ... + $n_{m,i})\rangle$'-resilient with respect to a particular security objective. This value describes a tuple of tuples, while the i^{th} tuple describes the i^{th} possibility to violate a particular security objective. The corresponding value for the small example is $\langle(A + B$ out of $A + B)$, $(C + D$ out of $C + D)\rangle$.

Some k-resilience Values for the Two Anaylsed Systems: In the Estonian system the set of possible intruders on the voting server side[4] are

- the manufactory,
- the administrators (from the authors knowledge, there are two administrators - admin$_1$ and admin$_2$ – with the same rights with respect to the server administration), and

Based on the architecture described in Sect. 9.2.1 and the possible intruders, the following exemplary values for the secrecy of the vote aspect of O.T.TamperServer can be computed:

- 1 out of 1 with respect to the manufactory
- 1 out of 2 with respect to the administrators

This can be combined to one value: $\langle$(1 out of 1), (1 out of 2)$\rangle$. Other combinations, for instance where the manufactory and the poll workers maliciously cooperate do not need to be taken into account because they do not constitute an additional possibility to manipulate.

In POLYAS system the set of possible intruders on the voting server side are

[4] For simplification reasons the voter as a possible intruder and outside intruders are not discussed in this part.

- the manufactory,
- the administrators (there are three administrators - admin$_{ERS}$, admin$_{VS}$, and admin$_{BBS}$ – while each of them hosts one of the three servers.

Based on the architecture described in Sect. 9.3.1 and the possible intruders, the following exemplary values for the secrecy of the vote aspect of O.T.TamperServer can be computed:

- 1 out of 1 with respect to the manufactory
- 1 + 1 out of 1 + 2 with respect to the administrators (as either admin$_{ERS}$ or admin$_{VS}$ needs to maliciously cooperate with admin$_{BBS}$)

This can be combined to one value: $\langle(1 \text{ out of } 1), (1+1 \text{ out of } 1+2)\rangle$. Note, this value would change if the computation would not address the secrecy of the vote aspect of O.T.TamperServer but the modification of votes aspect.

This k-resilience value show that the POLYAS system implements better the separation of duty principle on the architecture level than the Estonian system does.

10.3 Summary

This chapter focuses on the third aspect of the trust model – who can be trusted not to maliciously cooperate with others. This aspect is not covered by the presented Protection Profile and generally cannot be integrated in the Protection Profile.

Section 10.1 explains the motivation and importance of the integration of this third aspect of the trust model definition. It is not enough for a particular system to be certified against the framework from Chap. 8 but the responsible election authority has also to agree on this aspect of the trust model and argue why this third aspect holds for their election and environment. Having this information, the responsible election authority is able to respond with additional organisational measures, if necessary; like in the Estonian case where the administrators were observed by an auditor and a cameraman as well as policeman to ensure that no one other than this group of four people (in total) get physical access to the voting servers.

In Sect. 10.2, the k-resilience approach is presented, to determine the set of entities which needs to be trusted not to cooperate. These entities can be inside or outside intruders as well as the developers. Moreover, this value can be different from requirement to requirement. These different possibilities are shown for two examples: the Estonian system and the POLYAS system.

According to the values for the two systems under consideration, it is shown that even when both systems can get a Common Criteria certificate (for their compliance to the GI/BSI/DFKI Protection Profile), their k-resilience value can differ. Therefore, this value reveals additional information necessary to decide whether a particular system is secure enough or not.

Therefore, the combination of the evaluation according the Protection Profile and the calculation of the k-resilience value together with a judgment that the complete trust model meets the environment in which the system will be used, provides the basis for secure remote electronic elections.

11

Future Work - Open Issues

Even if the presented evaluation framework is an important step to provide a trustworthy basis for secure electronic voting, much work still needs to be done. The most important aspects are presented in this chapter.

Experiences form First Evaluations. The presented framework in terms of the extended framework in terms of the improved and extended Protection Profile from Chap. 8 is a good starting point but might need to be edited and/or extended after having evaluated the first systems against this PP. There are currently two German companies planning to evaluate their systems: Micromata and T-Systems. For such evaluations it is important to define the TOE in a way that it is flexible usable for different kind of elections. For instance, the ballot display and configuration should be outside the scope of the TOE. This is an important task for the Security Target author.

Covering Requirements from Other Catalogues. In this book, the coverage of requirements from the following three existing catalogues has been shown:

- The German Regulations for Electronic Voting Machines [143]
- The recommendations of the Council of Europe [37]
- The "Online-Voting Systems for Non-parliamentary Election" catalogue developed by the Physikalisch-Technische Bundesanstalt (PTB - Department of Metrological Information Technology in the National Metrology Institute) [62].

However, to confirm the completeness notion, the remaining literature discussed in Chap. 3 needs to be assigned to the presented list of requirements.

Remote Electronic Voting Framework Extension. The current framework in terms of the core Protection Profile only addresses remote electronic voting systems; in addition, only the security functions for the polling phase and the tallying phase are addressed. However, the election setup phase is equally important to ensure the election principles. If problems appear in this phase,

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 203–206, 2009.

trust in the electronic voting system will disappear in general, even if evaluators argue that everything is fine, because the evaluated part of the system does not support the election setup phase.

Section 8.3 addresses that error handling and error recovery should be further discussed. Resulting more concrete organisational security policies should be adapted to the Protection Profile.

Other functionality, which is not yet discussed in this book, but which should be addressed in such a framework, in future, is the application of vote updating as used in Estonia as well as updating the electoral register. Other important functionalities, which should be discussed in the future, are the resistance against disputations and different types of verification techniques (see Sect. 4.5).

Framework for Other Types of Election Systems. The evaluation framework, in terms of the core Protection Profile, the discussion of different trust models, as well as the discussion of different evaluation assurance levels are only done for remote electronic voting. Such a core Protection Profile needs to be developed for any type of electronic voting system identified in Sect. 2.1. For stand-alone direct recording electronic voting machines, the first step is already taken with Chap. 5 where corresponding requirements are already defined. As a first step, these requirements should be integrated into a core Protection Profile. Next, the focus should be on paper-based, electronic voting systems. Here, the work and experience from the Protection Profile for the Digital Election Pen [158] should be borne in mind.

Evaluation Methodology for Remaining Requirements. Section 8.3 already addresses the necessary discussion, whether to include all functional requirements provided in Sect. 5.4 and Sect. 6.4 respectively in a Protection Profile or to shift those functional requirements which only address the correct implementation of a functional requirement to an additional evaluation using other techniques. Here, an expert discussion is necessary where the advantages and disadvantages of both approaches are discussed and finally they need to come to a decision and recommend one of these two approaches. Depending on this decision, these remaining functional requirements need to be included in the Protection Profile approach or they need to be instantiated in the other approach.

Beside this discussion, evaluation standards for usability and organisational requirements need to be defined:

Usability System Requirements. Most of the usability requirements are not electronic voting specific ones. In general, one wants to have user-friendly systems and interfaces. Many of the development techniques and guidelines are available for usability engineering (such as [26]), usability testing (such as thinking aloud, constructive interaction, and questionnaires), and usability evaluation (such as heuristic evaluation, cognitive walkthroughs, formal usability inspections, pluralistic walkthroughs, and consistency inspection) [134]. However, it is an important aspect; thus, a concrete evaluation methodology needs to be defined here in order to get impartial, comparable, and repeatable evaluation results for the usability requirements as well.

Organisational Requirements. Organisational requirements together with the assumptions about the environment (aspect 1 from the trust model) are more than guidelines for the responsible election authorities : they are very important pre-conditions. If these pre-conditions are not met, the certification for the electronic voting system is lapsed and, thus, no statement about the compliance with the requirements and with the election principles can be made. Therefore, it is necessary to check that these requirements and assumptions about the environment are met by the responsible election authorities . Thus, corresponding evaluation methodologies need to be defined and demanded. One possibility may be the application of election observation as described for traditional elections in [110]. However, there is still research to be done to extend this handbook for electronic voting (see, for example, [88,89] and [119]).

Extension of the Formal Security Model. Section 7.4 provides an extract of a formal security model for remote electronic voting systems in order to enable the evaluation against high evaluation assurance levels. This extract should be completed for all security and functional requirements in the future. Some researcher also started to analyse electronic voting systems with formal methods: for instance, the voting protocol from Fujioka, Okamoto, and Ohta proposed in [48] is analysed in [98] by means of an ACP style process algebra, and in [83] by means of the applied pi-calculus. These approaches should be integrated into an extended formal IT security model.

Supporting the Responsible Election Authority. The title of this book claims to provide a standardised evaluation as a decision base for the responsible election authority. Indeed, based on the provided evaluation framework, the responsible election authority can decide whether a particular electronic voting system is in accordance with their needs. However, to decide this, they need to define the trust model, which holds for their election. This is a rather difficult task for people with a non-technical background. So, in the future, in order to apply this framework, a guideline for the responsible election authority discussion possible trust model and their differences are essential.

Requirement Integration into Election Laws. This book bases on a mainly technical driven discussion. This holds in particular for the definition of the trust model in the security problem definition part and for the selected evaluation assurance level. However, the lawyers are the one who decided for a particular election whether the chosen trust model fits to low level elections or whether even less requirements are necessary or vice versa that more requirements are necessary even for elections on low levels like in societies. The identified changes needs to be added in the framework.

Then, once having such a framework that is accepted by the lawyers, the next question is, how to integrate the framework into the election regulations. First of all, the general effort to change election laws and regulations depends on the election at hand. The city of Hamburg demanded the compliance to their own Protection Profile in the election regulations (see [158] and Sect. 3.1.2). The Hamburg approach is a singular example because they charged the development of a Protection Profile for the Digital Election Pen. Thus,this Protection Profile does not take any other form of electronic voting systems into account. In contrast, this book proposes to have a core Protection Profile as a basis to develop extended PPs for high level elections (addressing another trust model and demanding a higher evaluation assurance level). Here, lawyers need to check whether corresponding extended Protection Profiles must be developed and certified or whether a reference to the core Protection Profile together with the description of the trust model and evaluation assurance level for the corresponding election is sufficient, and then a decision can be made regarding the inclusion of Protection Profiles in the law.

In addition, new laws need to be enacted to to criminalise behaviours such as coercion of the voter, hacking voting systems or individual votes, jamming a voting system or preventing access to the system.

Trust and Transparency. Having an electronic voting system, which is evaluated against a high evaluation assurance level and with a minimised trust model, does not necessarily imply that the voters trust the system. The voter, himself, can often not decide whether to trust the chosen electronic voting system or not due to his non-technical background. He may listen to the public and the press. Thus, to introduce electronic voting successfully an elaborate information campaign is important. The campaign should explain to the voter, but also to the poll workers, the technical aspects from a high level point of view as well as inform them about the undertaken evaluation and certification steps. For such a campaign, it is important to consider the corresponding operational requirements from Sect. 6.6.2 (mainly Op.2, Op.5, Op.6, Op.8, Op.9, Op. 10, and Op.15). One very important aspect with respect to transparency is Assur.5, which demands the disclosure of all technical information[1]. Besides these requirements, the topic handling of negative press also needs to be addressed. The development of concepts for such a campaign is a very important aspect of the future work.

[1] Note, security by obscurity is a faulty idea for the application of electronic voting.

Part V

Conclusion

12

Summary and Concluding Words

The goal of this book is to develop a trustworthy base for electronic voting by providing an evaluation and certification framework based on

- a standardised, consistent, and exhaustive list of security, functional, usability, organisational, and assurance requirements, and
- a standardised evaluation methodology which
  - supports the defined security, functional, and assurance requirements,
  - is flexible with respect to different trust models (including assumptions for the environment, the intruder's technical capability, and entities that do not maliciously cooperate),
  - is flexible regarding different evaluation depths, and
  - produces impartial, comparable, and repeatable evaluation results by providing a guideline for the evaluator.

First, the *fundamentals* are discussed in order to be able to develop such a framework. This includes a general introduction to electronic voting and a classification of different election forms according to the medium in use, the environment where people cast their vote, and the point in time when vote casting is enabled. Moreover, a description and analysis of different implementations of electronic voting systems is provided, which concentrates on stand-alone direct recording electronic voting machines in polling stations and remote electronic voting systems. This includes a description and discussion of different implementations of the voter authentication technique, the way the secrecy of the vote is ensured, and the used client-side voting software. In addition, the fundamentals contain a discussion and analysis of existing approaches for the evaluation of electronic voting systems. Here, the vulnerabilities of existing documents are identified.

In order to provide a *standardised and consistent* list of requirements based on these fundamentals, corresponding syntax and semantics and a glossary for (electronic) voting terminology are provided and applied for the requirement specification. Furthermore, some of the specified security and functional requirements are defined in a formal IT security model. This model provides

M. Volkamer: Evaluation of Electronic Voting, LNBIP 30, pp. 209–213, 2009.

an unambiguous interpretation of the considered requirements in a formal language. To ensure that the list is *exhaustive*, it is verified that all requirements from existing literature are taken into account[1]. Moreover, it is checked whether all aspects of the election principles are addressed and all identified threats are covered. *Consistency and exhaustiveness* are further emphasised by the security rationale part of the GI/BSI/DFKI Protection Profile, which shows that the list of security requirements matches the security problem definition and, thus, the underlying trust model.

The results of this part are two lists of requirements containing system requirements (divided into functional, security, and usability requirements), organisational requirements, and assurance requirements – one for stand-alone direct recording electronic voting machines in polling stations and one for remote electronic voting systems.

The provided *evaluation and certification framework* is based on the Common Criteria (CC) evaluation standard, which exactly provides the required properties for the evaluation methodology: the security, functional, and assurance requirements are supported by the CC. In addition, the Common Criteria supports different trust models in terms of the security problem definition and the four attacker potential values: basic, enhanced-basic, moderate, and high. The Common Criteria together with the corresponding Common Evaluation Methodology (CEM) strictly guides the evaluator in a way that the evaluation results aim to be impartial, comparable, and repeatable. In addition to the required properties, the CC is an international standard, that is, remote electronic voting systems which are certified in one country can use the certificate in any other country that uses or accepts the Common Criteria without any additional effort. Furthermore, the CC features a certification process, which also monitors the quality of the evaluation and administers the regulations to which the evaluation facilities and evaluators must conform.

To apply the Common Criteria for electronic voting systems, the following challenges are taken:

- The defined security, functional, and assurance requirements are translated into the Common Criteria framework.
- Two different trust models, in terms of the CC and their consequences for the system design, are discussed: the temporary unlimited secrecy of the vote and the trustworthiness of the vote-casting device.
- A first step to develop a formal IT security model is taken. Such a model would be the base for a Common Criteria evaluation according to EAL 6 or 7.

According to the Common Criteria, a Protection Profile (PP) – describing specific implemen-tation-independent statements of security needs – is developed. This is only elaborated for remote electronic voting systems. However, having defined the PP for remote electronic voting systems, it can easily be

[1] For those requirements that have been left out, a chapter for a statement is given.

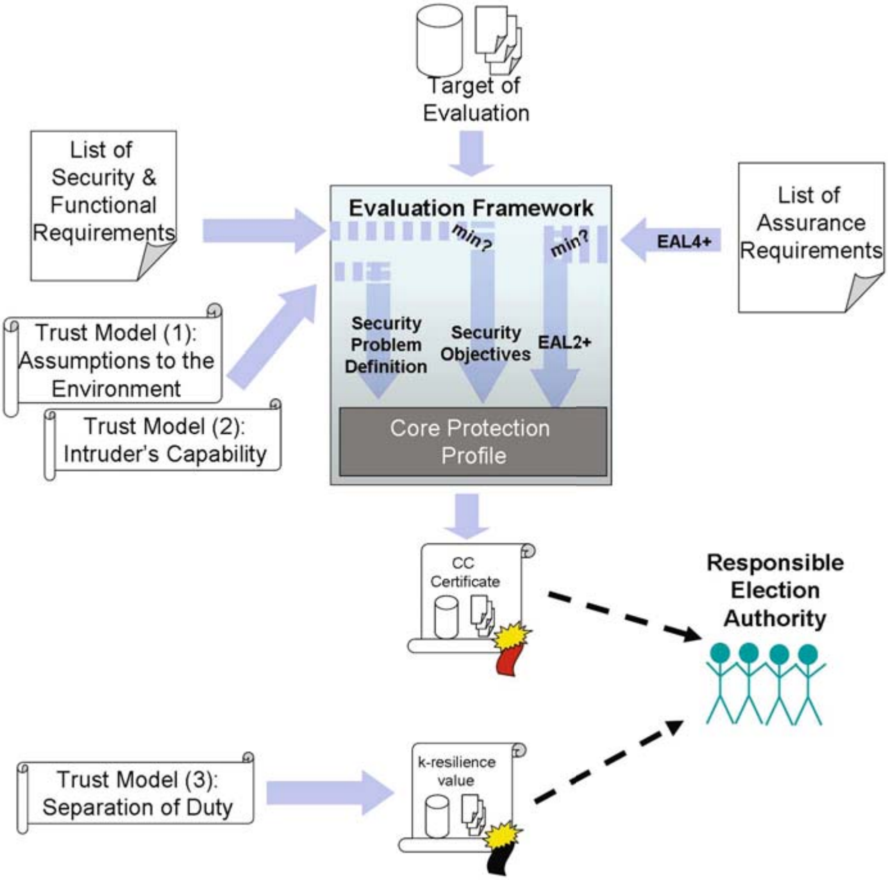

Target of Evaluation
List of Security & Functional Requirements
Evaluation Framework
min?
min?
EAL4+
List of Assurance Requirements
Trust Model (1): Assumptions to the Environment
Trust Model (2): Intruder's Capability
Security Problem Definition
Security Objectives
EAL2+
Core Protection Profile
CC Certificate
Responsible Election Authority
Trust Model (3): Separation of Duty
k-resilience value

Fig. 12.1. Framework overview

adapted and/or extended for electronic voting machines or other forms of electronic voting. As the required evaluation depth and the underlying trust model differs from election to election, it is argued that a core Protection Profile is essential, which can be extended in arbitrary ways, but which needs to be ensured by all remote electronic voting systems independent for which elections they are used. The provided core Protection Profile is based on an improvement of the GI/BSI/DFKI Protection Profile. One main property of this Protection Profile is that it subsumed different voting servers to one voting server to be applicable to as many voting protocols as possible.

In a last step, the Protection Profile is analysed with respect to its *applicability* for the praxis. Thus, in terms of a proof of concept, two available remote electronic voting systems, namely the POLYAS system and the Estonian system, are evaluated against the improved Protection Profile (in terms of the security objectives). The results of this analysis show that the Common Criteria is in general suitable for remote electronic voting systems as both systems could generally pass a formal Common Criteria evaluation. However, the analysis also shows that the proposed core Protection Profile does not handle all aspects of the trust model. It mainly focuses on the definition of assumptions about the environment and the intruder's technical capabilities, while the separation of duty approach is left open. This results from the simplification of the system architecture to one voting server. The importance of the separation of duty principle is highlighted in general as well as in particular for electronic voting systems (already traditional elections implement the separation of duty principle; thus, this should be translated in the electronic word). Therefore, a new approach to handle this principle is presented: the computation of the k-resilience value, describing those entities who need to be trusted not to maliciously cooperate.

An overview of the developed framework is illustrated in Fig. 12.1: Based on both evaluation and certification results (a Common Criteria certificate – for a particular trust model and with a particular evaluation depth – and the k-resilience value), the responsible election authority has to decide whether the underlying trust model matches the situation in which they want to use the particular system.

The provided framework is suitable for many (primary) elections and serves as an essential starting point for further investigation to a universal application. Moreover, it is the first requirement and evaluation framework which makes the underlying trust model and the evaluation depth transparent to the responsible election authority. Finally, this framework serves as a contribution to the international (research) community and can be used as basis for further extensions to meet also other type of elections with a smaller trust model.

Implication for the Practice - Discussion of the Trust Model.

Deciding about trust models is a rather *new task* for the responsible election authority. The advantage of making the trust model explicit is, that the responsible election authority is now able to take organisational measurements to ensure that the corresponding assumptions about the voting environment are met in their situation. The enforcement of such an evaluation is a legal question, which is outside the scope of this book.

Implication for the Practice - Confidence for the Responsible Election Authority and the Voters.

This framework allows the responsible election authorities and the voters to gain confidence, that a particular electronic voting system can be used for their election in a particular environment. This effect originate from:

- The confidence that the requirements are exhaustive. This is established by working with and improving existing voting requirement literature, by providing an approach for a formal IT security model to check the security and functional requirements, as well as by the development of a Protection Profile where it is shown that the requirements fit the problem definition containing the underlying trust model.

- The confidence in the correct implementation of the security functions of the electronic voting system and the sufficiency to ensure the defined requirements under the specified trust model. This is provided by a Common Criteria evaluation, while the amount of confidence depends on the evaluation depth.
- The confidence that the chosen electronic voting system goes with the election environment it is used in. This is established by making the trust model explicitly (in all three aspects).

Part VI

Appendix

A

List of Acronyms

BSI	Bundesamt für Sicherheit in der Informationstechnik
CC	Common Criteria
CEM	Common Evaluation Methodology
CM	Content Management
DRE	Direct Recording Electronic
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz
EAL	Evaluation Assurance Level
GI	Gesellschaft für Informatik
HSM	Hardware Security Module
OSP	Organisational Security Policy
PP	Protection Profile
PTB	Physikalisch-Technische Bundesanstalt
SAR	Security Assurance Requirements
SFR	Security Functional Requirements
SSL	Secure Socket Layer
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

B

Links

All links have been checked and were working on 04-20-2008.

B.1 Electronic Voting Systems

Examples for electronic voting machines:

- Nedap (The Netherlands, Germany, USA)
  http://www.election-systems.eu/website/Read.php
- Diebold Accu-Vote (USA)
  http://www.diebold.com/
- Election Systems & Software - ES&S (USA)
  http://www.essvote.com/
- Sequoia (USA)
  http://www.sequoiavote.com/
- Guardian Voting Systems (USA)
  http://guardianvoting.com/gvs/
- Advanced Voting Solutions (USA)
  http://www.advancedvoting.com/
- INDRA (Spain)
  http://www.indra.es/
- Digital Election Pen
  http://www.dotvote.de/

Examples for remote electronic voting systems:

- POLYAS
  http://www.micromata.de/produkte/polyas.jsp
- RIES
  http://www.rijnland.net/ries (available at June 11th 2008)
- Scytl
  http://www.scytl.com/

- everyone*counts*
  http://www.everyonecounts.com
- SENSUS
  http://lorrie.cranor.org/voting/sensus
- EVOX
  http://groups.csail.mit.edu/cis/voting/voting.html

B.2 Electronic Voting Antagonists

Electronic voting antagonists in Europe are:

- Europe: Europeans For Verifiable Elections
  http://www.efve.eu/ (available at June 11th 2008)
- Belgium: Pour une Ethique du Vote Automatisé
  http://www.poureva.be/
- France: Recul Democratique
  http://www.ordinateurs-de-vote.org/
- Ireland: Irish Citizens for Trustworthy Evoting
  http://evoting.cs.may.ie/
- Italy: Italien Electronic Voting and Democracy
  http://www.electronic-vote.org/
- The Netherlands: Wij vertrouwen stemcomputers niet
  http://www.wijvertrouwenstemcomputersniet.nl
- Germany: Chaos Computer Club
  https://www.ccc.de/

Electronic voting antagonists in the U.S. are:

- Black Box Voting
  http://www.blackboxvoting.org/
- Electronic Frontier Foundation
  http://www.eff.org/
- Open Voting Consortium
  http://openvotingconsortium.org/

C

Glossary

Election terminology is prone to ambiguity. Words may be used as both a verb and a noun (for example, 'vote'); some words can have subtly different meanings (for example, 'ballot' as piece of paper or as set of voting options or even as the choice of candidates). Therefore, the following definitions have been developed to provide unambiguous meanings to terms that are used in this book. Most of these items have already been published in the glossary of [156]. Minor changes have been made. In addition, those items relevant for remote electronic voting have been added. Some items in the subsequent glossary only appear in the context of stand-alone direct recording electronic voting machines or only in the context of remote electronic voting. These are adequately labelled with (stand-alone direct recording electronic voting machine) and (remote electronic voting), respectively.

C.1 Election Terminology

In this section the general election terms are defined. To clarify the terms, some are illustrated in Fig. C.1.

ballot : ·*voting options*· presented on a form (as paper sheet or displayed on a screen)

ballot box : physical box in which tangible ·*votes*· (usually paper records) are stored

cast *(verb)* a vote(s)[1] : to finally and irrevocably commit a ·*vote(s)*·

cast vote : ·*vote*·, which is stored in the e-ballot box

cancel the voting process : to leave the ·*voting process*· without having cast a ·*vote*·

[1] In some circumstances more than one ·*poll*·/·*election*· run in parallel and, thus, the ·*voter*· ·*casts*· more than one ·*vote*·. Shareholder elections are another example where the number of ·*votes*· a ·*voter*· can ·*cast*· depends on the amount of shares.

election : the proceedings accompanying the formal choosing of the winner(s) of one or more ·*polls*·

electoral register : list of all ·*eligible voters'*· details necessary to unambiguously identify and authenticate ·*eligible voters*·. These details can, for example, be the ·*voter's*· name or his membership number as well as the information whether he has already ·*cast*· a ·*vote*·.

poll : a decision between ·*voting options*·, which is determined by ·*eligible voters*· ·*casting*· ·*votes*·

(election) result : election result of ·*vote*· tallying (includes the proper handling of valid and ·*spoilt votes*·)

selection : an indication by a ·*voter*· of some subset of ·*voting options*·

spoil : to ·*cast*· a ·*vote*·, which will not be counted for some legitimate reason, for example, incorrectly filled-in or blank

vote ***(noun)*** : the expression of an individual ·*voter's*· choice

voting option : a candidate, a party, a issue, or simply "yes/no" (in the case of a referendum)

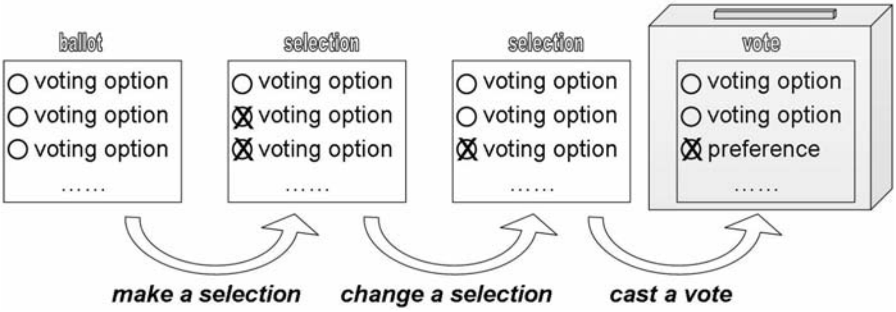


Fig. C.1. Ballot - voting option - selection -preference - vote

C.2 Electronic Voting Specific Terms

authentication data : data stored in the ·*electoral register*· used to authenticate ·*voters*·

authentication information : the data send by the voter to authenticate himself against the ·*electronic voting system*·

authentication token[2] : ·*voter's*· token used to authenticate himself against the ·*electronic voting system*·

ballot data : a set of information containing a list of ·*voting options*·, ·*ballot*· design information, and the definition of valid and ·*spoilt votes*·

confirmation : the ·*voter*· gets back a confirmation from the ·*electronic voting system*· if the ·*electronic voting system*· has successfully stored the ·*e-vote*·

e-ballot box : memory component of the ·*electronic voting system*· in which ·*e-votes*· are stored during the ·*polling phase*·

e-vote : an electronic record of a ·*vote*·

election data : those data store on the ·*electronic voting system*· after the ·*tallying phase*· (at least including: ·*votes*·, ·*results*· and audit data)

identification data : data stored in the ·*electoral register*· to identify ·*voter*·. This data is also used by the ·*voter*· as his identification token

polling phase data : those data store on the ·*electronic voting system*· when starting the ·*polling phase*· (at least including: ·*ballot data*· and ·*electoral register*·)

C.3 Phases of the Election

In this section the different phases of any election are described. To clarify the terms, some are illustrated in figure C.2, C.3 and C.4.

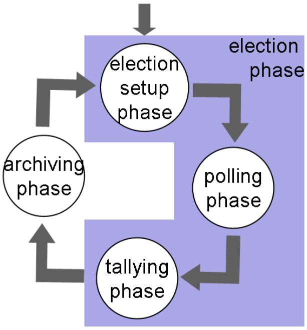


Fig. C.2. Election circle

[2] In the case of secret based authentication technique, the authentication token would be the TAN itself while the authentication information and authentication data could be the encrypted TAN.

active state (stand-alone direct recordingelectronic voting machine) : state in which *·votes·* can be *·cast·* (or the *·voting process·* canceled) at the *·electronic voting machine·*. The *·electronic voting machine·* is in an *·active state·* for the duration of the *·voting process·*

archiving phase : phase after the *·election phase·* for which *·vote·* records must be retained

election phase : phase from the beginning of the *·election setup·*, through the *·polling phase·*, to the completion of the *·tallying phase·*

election setup : phase in which the *·electronic voting system·* is prepared, configured and distributed. This also includes the organisational preparation. In the case of remote electronic voting this phase includes the preparation of the *·electoral register·*, moreover, if necessary, the generation and distribution of authentication tokens to the *·voter·*

inactive state (stand-alone direct recording electronic voting machine) : state in which no *·votes·* can be *·cast·* at the *·electronic voting machine·* (nor the *·voting process·* canceled)

polling phase : phase of time when polls are open, that is *·votes·* can be *·cast·* and corresponding *·e-votes·* are stored in the *·electronic voting system·*

tallying phase : phase of result calculation. This may entail the opening of the collected *·votes·* and the result calculation based on this opened *·votes·*

voting process : all interactions of an *·eligible voter·* with the electronic voting system. In the case of *·electronic voting machine·*, the *·voting process·* begins when the *·electronic voting machine·* is put in an *·active state·*. In the case of remote electronic voting the *·voting process·* begins when an *·eligible voter·* has been identified and authenticated as such. In both cases, it ends when the voter has cast his *·vote(s)·* or cancels the *·voting process·*

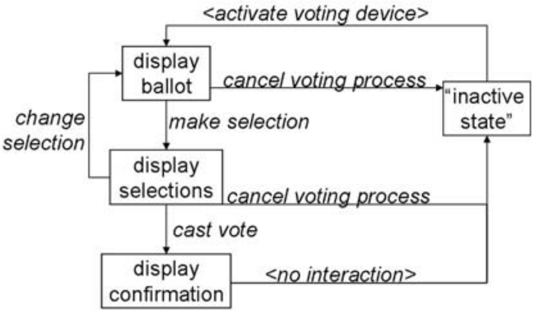


Fig. C.3. Polling phase with electronic voting machine in detail

C.4 Participants

In this section the people and groups involved in an election (especially those involved in an election where an electronic voting system is used) are defined.

administrators : sub group of ·*poll workers*·. They technically assist the traditional ·*poll workers*·

certification authority : body (or bodies), which certifies the ·*electronic voting system’*· compliance with requirements as a whole or partially

elector : ·*voter*· who has ·*cast*· his ·*vote*·

eligible voter : ·*voter*· who has not ·*cast*· his ·*vote*·, yet

ineligible voter : a person who tries to ·*cast a vote*· but who is not listed in the ·*electoral register*·

manufacturer : body (or bodies) responsible for the development and maintenance of ·*electronic voting system*· as a whole or partially

poll worker : a person in his role as an official facilitator in the running of an ·*election*·, like government employees or citizen panels. Technical assistants are included

responsible election authority : the organisation responsible for running the ·*poll*·/·*election*·

testing authority : body (or bodies), which tests the ·*electronic voting system’*· compliance with requirements as a whole or partially

user : anyone who is authorised to interact with an ·*electronic voting system*· during the ·*polling phase*·, ie, ·*poll workers*· and ·*eligible voters*·

voter : person listed in the ·*electoral register*·

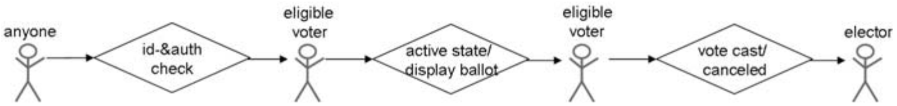


Fig. C.4. Anyone - eligible voter - voter elector

C.5 Devices and Components

Different devices are necessary to run an election. These devices contain different components. Both devices and components are defined in this section.

audit system : sub-system of the ·*electronic voting system*· which allows the actual behaviour of the ·*electronic voting system*· to be observed

ballot box server (remote electronic voting) : component of a remote electronic voting system which stores the ·*e-votes*·. It is part of the ·*voting server*· and can be divided into different ·*ballot box servers*·

client-side voting software (remote electronic voting) : application running on the ·*vote-casting device*· used by ·*eligible voters*· to ·*cast*· their ·*vote(s)*·

tallying phase : Process which starts after the end of the polling phase and works on the ·*e-votes*·. The election result of the tallying phase is the ·*election result*·

tallying software : component of the ·*electronic voting system*· which calculates ·*poll*·/·*election*· results

electronic voting system : any software and hardware component as well as infrastructure involved in the ·*poll*·/·*election*·

poll worker interface : user-interface to the ·*electronic voting system*· which enables ·*poll workers*· to carry out their duties

server-side voting software (remote electronic voting) : application running on the ·*voting server*· used to run the ·*polling phase*·

vote-casting interface : user-interface for the ·*voter*· in the ·*voting process*·

vote-casting device (remote electronic voting) : component used by the ·*voter*· to communicate with the ·*voting server*·

voting channel : a medium through which ·*voters*· can ·*cast*· ·*votes*·

electronic voting machine (stand-alone direct recording electronic voting machine) : device to ·*cast*· ·*voters*· and store corresponding ·*e-votes*·

voting server (remote electronic voting) : central component of the ·*electronic voting system*·. In general it is a high level abstraction of different single voting servers

C.6 Assessing Terminology

According to [162] assessing related terminology is defined in the following way:

audit : a systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled (ISO 9000:2000). It is, in other words, a verification process intending to verify that a quality management system meets requirements (requirements as defined, for example, by ISO 9001:2000). It is the processes rather than the products that are verified.

observation : A process including elements of both review and audit, with a main focus on verifying election processes based upon samples of evidence.

certification : A process performed by an independent organisation (certification body) where a quality management system or a product is verified (or validated) to demonstrate through objective evidence that specified requirements (for example, ISO 9001:2000 for quality management systems) are met.

C.7 Mapping: PP Glossary – Book Glossary

The comparison in Sect. 8.3 bases on a glossary mapping from Sect. C [161] to [160] respectively. These are provided in Table C.1 and C.2 (the latter one discusses only those entries where different English words are used).

Table C.1. Mapping words from [161]'s to those from appendix C

GI/BSI/DFKI PP	Book
Abgegebene Stimme	cast vote
Authentisierungsdaten	authentication data
Authentisierungsmerkmal	authentication token
Clientseitiger EVG	client-side voting software
Endgerät	vote-casting device
(Wahl-)Ergebnis	(election) result
Identifikationsdaten	identification data
Registrierter Wähler	voter
Rückmeldung	confirmation
Serverseitiger EVG	server-side voting software
Stimmabgabe	casting a vote
Stimmauszählung	tallying
Stimmdatensatz	e-vote
Stimme	vote
Stimmzettel	ballot
Stimmzetteldaten	ballot data
Unbefugter Wähler	ineligible voter
Urne	e-ballot box
Wähler	voter + ineligible voter
Wähler mit Stimmberechtigung	eligible voter
Wähler ohne Stimmberechtigung	elector
Wahlberechtigungsliste	electoral register
Wahldaten	polling phase data
Wahldurchführung	polling phase
Wahldurchführungsdaten	election data
Wahlhandlung	polling process
Wahlserver	voting server
Wahlveranstalter	responsible election authority
Wahlvorschläge	voting options
Wahlvorstand	poll worker
Zischenergebnis	intermediate result

For the following items from the German version of the BSI/GI/DFKI Protection Profile, no corresponding words are defined in appendix C: Inhalt der Authentisierungsnachricht, Separation of Duty, Stimmabgabevermerk, Wahlende, Wahlende-Zeitpunkt, Wahlgeheimnis, Zugang, Zugriff, Zutritt, Zwischenspeicher.

Table C.2. Mapping words from [160]'s to those from appendix C

GI/BSI/DFKI PP (English)	Book
election officers	poll workers
vote record	e-vote
acknowledgement	confirmation
registered voter	voter
unauthorised voter	ineligible voter
voter without the right to vote	elector
election server	voting server
abort	cancel
polling process	voting process
election officers	poll worker
secrecy of voting	secrecy of the vote
ballot box	e-ballot box

D

Removed Requirements

There are different categories of reasons why particular requirements have been left out. The different categories are listed together with the assigned requirements.

Registration

The following requirements have been removed because they address the process to register as a voter to cast an e-vote. As this is part of the election setup phase which is not regarded, these do not need to be considered:

CoE 2 “Possible registration requirements for e-voting shall not pose an impediment to the voter participating in e-voting.

CoE 40 “The possibility of creating an electronic register and introducing a mechanism allowing online application for voter registration and, if applicable, for application to use e-voting, shall be considered. If participation in e-voting requires a separate application by the voter and/or additional steps, an electronic, and, where possible, interactive procedure shall be considered.”

CoE 41 “In cases where there is an overlap between the period for voter registration and the polling period, provision for appropriate voter authentication shall be made.”

CoE 88 “The fact that voter registration has happened within the prescribed time limits shall be ascertainable.”

Not Electronic Voting Specific

The following requirements have been removed because they are not specific ones for electronic voting but also hold for elections in general:

CoE 9 “The organisation of e-voting shall secure the free formation and expression of the voter’s opinion and, where required, the personal exercise of the right to vote.”

CoE 18 "The e-voting system shall be so designed that the expected number of votes in any electronic ballot box will not allow the result to be linked to individual voters."

CoE 39 "There shall be an electoral register which is regularly updated. The voter shall be able to check, as a minimum, the information which is held about him/her on the register, and request corrections."

CoE 54 "The e-voting system shall prevent processing information on votes cast within deliberately chosen sub-units that could reveal individual voters' choices."

PTB (PE 1-1) "The electronic register of voters shall correctly reflect the register of voters valid."

PTB(PE 4-7) "Appropriate precautions for the unobserved marking of ballots shall be taken at the polling station."

PTB(VP 3-7) "During ballot marking confidentiality shall be guaranteed."

BWGV (5) "Soweit sich aus den Vorschriften dieser Verordnung nicht etwas anderes ergibt, gelten auch bei der Verwendung von Wahlgeräten die Vorschriften der Bundeswahlordnung oder der Europawahlordnung."

BWGV (9-1) "Jedes Wahlgerät ist in der Wahlzelle so aufzustellen, daß jeder Wähler seine Stimmen unbeobachtet abgeben kann."

BWGV (9-2) "Die gerätespezifische Darstellung der Wahlvorschläge bei Bundestagswahlen ist so anzuordnen, daß sich die Wahlvorschläge für die Erststimmen vom Wähler aus gesehen links oder oben befinden."

BWGV (11-1) "Für die Stimmabgabe an den Wahlgeräten gelten die §§ 56 und 58 der Bundeswahlordnung oder die §§ 49 und 51 der Europawahlordnung mit den in den Absätzen 2 bis 5 genannten Maßgaben."

BWGV (11-2) "Nach Betreten des Wahlraumes begibt sich der Wähler an den Tisch des Wahlvorstandes und nennt seinen Namen. Dabei soll er die Wahlbenachrichtigung abgeben. Auf Verlangen hat er sich über seine Person auszuweisen."

BWGV (14-4) "Den abgegebenen ungültigen Erst- und Zweitstimmen (Absatz 3 Satz 1 Nr. 5) sind die in der Zählliste aufgeführten gemäß § 11 Abs. 4 Satz 3 ungültigen Stimmen hinzuzurechnen."

Over-Specification

Some requirements are too concrete (over-specified) and, thus, they would exclude potentially good electronic voting systems because they are not using a particular technology. Requirements assigned to this category are:

CoE 55 "'Any decoding required for the tallying of the votes shall be carried out as soon as practicable after the closure of the polling period."

CoE 108 "The audit system shall provide the ability to verify that an e-election or e-referendum has complied with the applicable legal provisions, the aim being to verify that the results are an accurate representation of the authentic votes."

PTB(VP 2-1a) "The electronic register of voters shall permit vote casting marks to be set."

PTB(VP 2-5) "The electronic register of voters shall be separated from vote storage."

PTB(VP 5-4) "As part of the vote storage process, feedback to the electronic register of voters shall take place."

BWGV-A1 A (c) "Selbsttätige Zählung der insgesamt abgegebenen Stimmen mit zugehöriger Anzeige."

BWGV-A1 A (d) "Selbsttätige Zählung der abgegebenen Stimmen sortiert nach den Wahlvor-schlägen bzw. nach ungültig gekennzeichneten Stimmen mit Anzeige des Zählergebnisses."

BWGV-A1 B (2.6a) "Das Wahlgerät kann gut transportiert [..] werden"

BWGV-A1 B (3.5b) "Vor Beginn der Wahl kann die Wirkung genau derjenigen Bedienungsvorrichtungen, die zur Auswahl der Stimmabgabe für einen der Wahlvorschläge nicht benötigt werden, für die Dauer des gesamten Wahlvorganges gesperrt werden."

Additional Functionality

The following requirements have been removed because they address functionality which is explicitly left out in the target of evaluation description:

CoE 42 "The possibility of introducing online candidate nomination may be considered."

CoE 43 "A list of candidates that is generated and made available electronically shall also be publicly available by other means."

CoE 49 "If it is decided that information about voting options will be accessible from the e-voting site, this information shall be presented with equality."

CoE 87 "The fact that candidate nomination and, if required, the decision of the candidate and/or the competent electoral authority to accept a nomination has happened within the prescribed time limits shall be ascertainable."

CoE 89a "The integrity of data communicated from the pre-voting stage (for instance electoral register and lists of candidates) shall be maintained."

PTB(VP 2-2) "Eligible short-term changes of the electronic register of voters (corrections and/or amendments) by authorised persons shall be allowed and easily manageable."

PTB(VP 2-4) "Any change of the electronic register of voters shall be logged."

PTB(CF 1-10) "The electronic register of voters must not be changed during an inter-ruption."

PTB(CF 1-13) "The system shall be set back to a state from which vote casting can be resumed."

Other Reasons

The following requirements have been removed for different reasons. These reasons are made explicitly per requirement:

CoE 27 "The e-voting system shall not prevent the partial or complete re-run of an election or a referendum."

Reason: If it is possible to re-run an election after having start the polling period, the poll worker might do so after a couple of cast votes (maybe from voters they do not like) which would mean that these e-votes are lost.

CoE 58 "In the event of any irregularity affecting the integrity of votes, the affected votes shall be recorded as such."

Reason: In general irregularities must not affect votes. Thus, no corresponding votes need to be recorded as such.

CoE 112 "In order to enhance international co-operation and avoid duplication of work, member states shall consider whether their respective agencies shall join, if they have not done so already, relevant international mutual recognition arrangements such as the European Cooperation for Accreditation (EA), the International Laboratory Accreditation Cooperation (ILAC), the International Accreditation Forum (IAF) and other bodies of a similar nature."

Reason: This requirement is lapsed because the book provides already one particular evaluation standard, namely the Common Criteria

PTB(VP 4-6b) "[..] all votes cast shall have been received by the vote storage."

Reason: This can not be guaranteed by any system because you never now whether a voter casts his vote in time but it takes some seconds to receive the voting server and thus, the poll is closed in the meantime.

BWGV (18) "Für Wahlgeräte einer Bauart, die bereits für die Wahlen zum 14. Deutschen Bundestag oder die Europawahlen 1994 zugelassen worden ist, gilt die Bauartzulassung im Rahmen des jeweiligen Zulassungserlasses des Bundesministeriums des Innern allgemein für Wahlen zum Deutschen Bundestag oder Europawahlen als erteilt. 2§ 8 Abs. 1 Nr. 6 ist auf diese Wahlgeräte nicht anzuwenden."

Reason: This requirement addresses the interim arrangement for already certified devices according to the old version of the regulation.

BWGV (19) "(weggefallen)"

Reason: This requirement is removed in the current version of the regulations.

BWGV (20) "Diese Verordnung tritt am Tage nach der Verkündung in Kraft."

Reason: This requirement determine the day when this regulation comes operative.

BWGV-A1 B (3.1d) “Bei getrennter Bedienung für Auswahl und Abgabe der Stimmen kann die Abgabe der Erst- und der Zweitstimme über eine gemeinsame Bedienungsvorrichtung erfolgen.”
Reason: This requirement is very specific for the German federal elections.

BWGV-A1 B (3.7b) “Bedienungshandlungen des Wählers ergeben keine Fehlermeldungen, sondern ggf. Hinweise zum Handlungsablauf. ”
Reason: Problems should be reported and displayed to the poll workers and not to the voter.

E

Protection Profile Structure

A Protection Profile contains six main parts and should be structured in the following way:

1. PP Introduction:
   - 1.1 PP reference
   - 1.2 TOE overview
2. Conformance Claim
   - 2.1 CC conformance claim
   - 2.2 PP claim, Package claim
   - 2.3 Conformance rationale
   - 2.4 Conformance statement
3. Security Problem Definition
   - 3.1 Threats
   - 3.2 Organisational security policies
   - 3.3 Assumptions
4. Security Objectives
   - 4.1 Security objectives for the TOE
   - 4.2 Security objectives for the operational environment
   - 4.3 Security objectives rationale
5. Extended Component Definition
6. Security Requirements
   - 6.1 Security functional requirements
   - 6.2 Security assurance requirements
   - 6.3 Security requirements rationale

References

1. Abe, M.: Universally Verifiable Mix-Net with Verification Work Independent of the Number of Mix-Servers. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 437–447. Springer, Heidelberg (1998)
2. Abe, M.: Mix-Networks on Permutation Networks. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 258–273. Springer, Heidelberg (1999)
3. Alkassar, A., Krimmer, R., Volkamer, M.: Online-Wahlen für Gremien – Wahlen in Gremien als Einsatzgebiet für Wahlen ohne vertrauenswürdige Instanz. Datenschutz und Datensicherheit – DuD (8), 480–482 (2005)
4. Alkassar, A., Sadeghi, A.R., Schultz, S., Volkamer, M.: Towards Trustworthy Online Voting. In: Proceedings of the 1st Benelux Workshop on Information and System Security – WISSec '06 (2006), https://www.cosic.esat.kuleuven.be/wissec2006/papers/17.pdf
5. Alkassar, A., Volkamer, M. (eds.): E-Voting and Identity – First International Conference, VOTE-ID 2007. LNCS, vol. 4896. Springer, Heidelberg (2007)
6. Andler, K.: Sicherheitsmechanismen eines Internetwahl-Systems außerhalb des Wahlprotokolls. Bachelor's thesis, Saarland University (2006)
7. Arzt-Mergemeier, J., Beiss, W., Steffens, T.: The Digital Voting Pen at the Hamburg Elections 2008: Electronic Voting Closest to Conventional Voting. In: Alkassar, A., Volkamer, M. (eds.) E-Voting and Identity – First International Conference, VOTE-ID 2007. LNCS, vol. 4896, pp. 88–98. Springer, Heidelberg (2007)
8. Baudron, O., Fouque, P.A., Pointcheval, D., Stern, J., Poupard, G.: Practical Multi-Candidate Election System. In: Proceedings of the 20th ACM Symposium on Principles of Distributed Computing – PODC '01, pp. 274–283. ACM Press, New York (2001)
9. Bederson, B.B., Lee, B., Sherman, R.M., Herrnson, P.S., Niemi, R.G.: Electronic Voting System Usability Issues. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '03, pp. 145–152. ACM Press, New York (2003)
10. Bell, D.E., LaPadula, L.: Secure computer systems: Mathematical foundation and a mathematical model. ESD-TR-73-278, MTR-2547, Vols 1&2, The MITRE Corporation, Bedford (1973)

11. Benaloh, J.: Verifiable Secret-Ballot Elections. PhD thesis, Yale University (1987)
12. Benaloh, J., Tuinstra, D.: Receipt-Free Secret-Ballot Election. In: Proceedings of the 26^{th} ACM Symposium on Theory of Computing – STOC '94, pp. 544–553. ACM Press, New York (1994)
13. Biba, K.J.: Integrity Considerations for Secure Computer Systems. Technical report, MITRE Corp., Bedford (1977)
14. BITKOM (Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.) and DIN (Deutsches Institut der Normierung e.V.: Kompass der IT-Sicherheitsstandards – Leitfaden und Nachschlagewerk., http://www.bitkom.org/files/documents/Kompass_der_IT_28.06.06.pdf
15. Braun, N.: E-Voting: Switzerland's Projects and their Legal Framework. In: Electronic Voting in Europe, pp. 43–52 (2004)
16. Braun, N.: Stimmgeheimnis – eine rechtsvergleichende und rechtshistorische Untersuchung unter Einbeziehung des geltenden Rechts. PhD thesis, Universität Bern (2006)
17. Bundesgesetz über den Schutz personenbezogener Daten – Datenschutzgesetz (DSG). Austria (2005), http://www.dsk.gv.at/dsg2000d.htm (last change 2005)
18. Bundesgesetz über die Kammern der gewerblichen Wirtschaft – Wirtschaftskammergesetz (WKG). Austria (2006), http://wko.at/reorg/organisationsrecht/wkg.pdf (last change 2006)
19. Bundesgesetz über die Vertretung der Studierenden 1998 – Hochschülerinnen- und Hochschülerschaftsgesetz (HSG). Austria (2005), http://www.oeh.ac.at/uploads/media/hsg2005reader_03.pdf (last change 2005)
20. Bundesgesetz über elektronische Signaturen – Signaturgesetz (SigG). Austria (2005), http://www.signatur.rtr.at/de/legal/sigg.html (last change 2005)
21. Bundesverfassungsgericht: BVerfG, 2 BvC 3/07 vom 03.03.2009, Absatz-Nr.(1-136) (2009), http://www.bverfg.de/entscheidungen/cs20090303_2bvc000307.html
22. Bundeswahlgesetz (BWahlG). Germany (2005), http://www.gesetze-im-internet.de/bundesrecht/bwahlg/gesamt.pdf (last change 17.03.2008)
23. Burmester, M., Magkos, E.: Towards Secure and Practical e-Elections in the New Era. In: Secure Electronic Voting. Advances in Information Security, pp. 63–76. Kluwer Academics Publishers, Dordrecht (2003)
24. Burton, C., Karunasekera, S., Harwood, A., Stanley, D., Ioannou, I.: A Distributed Network Architecture for Robust Internet Voting Systems. In: Wimmer, M.A., Traunmüller, R., Grönlund, Å., Andersen, K.V. (eds.) EGOV 2005. LNCS, vol. 3591, pp. 300–308. Springer, Heidelberg (2005)
25. Byrne, M.D., Greene, K.K., Everett, S.P.: Usability of Voting Systems: Baseline Data for Paper, Punch Cards, and Lever Machines. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '07, pp. 171–180. ACM Press, New York (2007)
26. Caldwell, B., Cooper, M., Reid, L.G., Vanderheiden, G.: Web Content Accessibility Guidelines 2.0 – W3C Working Draft (11 December 2007), http://www.w3.org/TR/WCAG20/
27. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Commununication of the ACM 24(2), 84–90 (1981)

28. Chaum, D.: Blind Signatures for Untraceable Payments. In: Advances in Cryptology – Proceedings of CRYPTO '82, pp. 199–203. Plenum Press, New York (1982)
29. Chaum, D.: Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 177–182. Springer, Heidelberg (1988)
30. Chaum, D.: Secret-Ballot Receipts: True Voter-Verifiable Elections. IEEE Security and Privacy 2, 38–47 (2004)
31. Chaum, D., Ryan, P.Y.A., Schneider, S.A.: A Practical Voter-Verifiable Election Scheme. In: de Capitani di Vimercati, S., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 118–139. Springer, Heidelberg (2005)
32. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In: Proceedings of the 26^{th} IEEE Symposium on Foundations of Computer Science – FOCS '85, Portland, pp. 383–395. IEEE Computer Society Press, Los Alamitos (1985)
33. Clark, D., Wilson, D.: A Comparison of Commercial and Military Security Policies. In: Proceedings of the 1987 IEEE Symposium on Security and Privacy, Washington DC, pp. 184–194. IEEE Computer Society Press, Los Alamitos (1987)
34. Cohen(Benaloh), J.D., Fischer, M.J.: A Robust and Verifiable Cryptographically Secure Election Scheme. In: Proceedings of the 26^{th} IEEE Symposium on Foundations of Computer Science – FOCS '85, Portland, pp. 372–382. IEEE Computer Society Press, Los Alamitos (1985)
35. Common Criteria for Information Technology Security Evaluation: Version 3.1 (2006), http://www.commoncriteriaportal.org/thecc.html
36. Common Evaluation Methodology for Information Technology Security Evaluation: Version 3.1 (2006), http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R2.pdf
37. Council of Europe: Legal, Operational and Technical Standards for E-Voting. Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe and explanatory memorandum. Council of Europe Publishing, http://www.coe.int/t/e/integrated_projects/democracy/02_activities/02_e
38. Cramer, R.J.F., Franklin, M.K., Schoenmakers, B., Yung, M.: Multi-authority Secret-Ballot Elections with Linear Work. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 72–83. Springer, Heidelberg (1996)
39. Cramer, R., Gennaro, R., Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)
40. Cranor, L.F.: In Search of the Perfect Voting Technology: No Easy Answer. In: Secure Electronic Voting. Advances in Information Security, pp. 17–30. Kluwer Academics Publishers, Dordrecht (2003)
41. Cranor, L.F., Cytron, R.K.: Sensus: A Security-Conscious Electronic Polling System for the Internet. In: Proceedings of the 30^{th} Hawaii International Conference on System Sciences – HICSS '97, Information System Track-Organizational Systems and Technology, vol.3, Washington DC, pp. 561–570. IEEE Computer Society, Los Alamitos (1997)
42. Dahl, R.A.: On Democracy. Yale University Press, London (1989)

43. DeMillo, R.A., Lynch, N.A., Merritt, M.Y.: Cryptographic Protocols. In: Proceedings of the 14^{th} ACM Symposium on Theory of Computing – STOC '82, pp. 383–400. ACM Press, New York (1982)
44. Organization for the Advancement of Structured Information Standards (OASIS). Technical report
45. Federal Election Commission: Voting System Standard. Agenda Documents 01-62 and 01-62a (2001), http://www.fec.gov/agenda/agendas2001/mtgdoc01-62/mtgdoc01-62.html
46. Office for Democratic Institutions and Human Rights, Kingdom of Belgium – Local Elections – 8 October 2006 – Expert Visit on New Voting Technologies (2006)
47. Forsgren, O., Tucholke, U., Levy, S., Brunessaux, S.: D4 Volume 3 Report on electronic democracy projects, legal issues of Internet voting and users (i.e. voters and authorities representatives) requirements analysis. Technical Report CYBERVOTE:WP2:D4V3:2001, EU CyberVote Project (2001), http://www.eucybervote.org/KISTA-WP2-D4V3-v1.0.pdf (11.06.2008)
48. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (1993)
49. Furukawa, J., Sako, K.: An Efficient Scheme for Proving a Shuffle. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 368–387. Springer, Heidelberg (2001)
50. Gesellschaft für Informatik: Ordnung der Wahlen und Abstimmungen (OWA) (2004), http://www.gi-ev.de/fileadmin/redaktion/OWA/gi-owa.pdf (last change 21.09.2004)
51. Gesellschaft für Informatik: Information für GI-Mitglieder zu möglichen Sicherheitsproblemen auf Clientseite bei Vorstands- und Präsidiumswahlen mit dem Onlinewahlverfahren (2007), https://www.gi-ev.de/fileadmin/redaktion/Wahlen/handreichungen_gi_onlinewahlen.pdf
52. Gesetz über die Wahl zur Hamburgischen Bürgerschaft. Germany (2007), http://hh.juris.de/hh/BuergWG_HA_1971_rahmen.htm (last change 20.12.2007)
53. Gesetz über Wahlen und Abstimmungen (Wahlgesetz). Switzerland (1994) http://www.gesetzessammlung.bs.ch/erlasse/132.100.pdf
54. Golle, P., Zhong, S., Boneh, D., Jakobsson, M., Juels, A.: Optimistic Mixing for Exit-Polls. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 451–465. Springer, Heidelberg (2002)
55. Grimm, R., Krimmer, R., Meißner, N., Reinhard, K., Volkamer, M., Weinand, M.: Security Requirements for Non-political Internet Voting. In: Krimmer, R. (ed.) Electronic Voting 2006, 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC. LNI, vol. 86, pp. 203–212. Gesellschaft für Informatik, Bonn (2006)
56. Grimm, R., Volkamer, M.: Development of a Formal IT-Security Model for Remote Electronic Voting Systems. In: Electronic Voting 2008. LNI, vol. 131, pp. 185–196. Gesellschaft für Informatik, Bonn (2008)
57. Grimm, R., Volkamer, M.: Entwicklung eines formalen IT-Sicherheitsmodells für Online-Wahlsysteme. In: Reduktion der Komplexität durch Recht und IT – IRIS '08 Tagungsband, pp. 145–156. Boorberg, Stuttgart (2008)
58. Grimm, R.: IT-Sicherheitsmodelle. WISU Das Wirtschaftsstudium, 720–727 (May 2008)

59. Grochulla, M.P.: Trust Model for eVoting. Bachelor's thesis, Saarland University (2007)
60. Groth, J.: A Verifiable Secret Shuffle of Homomorphic Encryptions. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 145–160. Springer, Heidelberg (2002)
61. Hammer, V., Pordesch, U., Roßnagel, A.: KORA – eine Methode zur Konkretisierung rechtlicher Anforderungen zu technischen Gestaltungsvorschlägen für Informations- und Kommunikationssysteme. Arbeitspapier 100, provet (1992)
62. Hartmann, V., Meissner, N., Richter, D.: Online Voting Systems for Non-parliamentary Elections – Catalogue of Requirements. Laborbericht PTB-8.5-2004-1, Physikalisch-Technische Bundesanstalt Braunscheig und Berlin, Fachbereich Metrologische Informationstechnik (2004), http://ib.ptb.de/8/85/LB8_5_2004_1AnfKat.pdf
63. Helbach, J., Krimmer, R., Meletiadou, A., Meißner, N., Volkamer, M.: Zukunft von Online-Wahlen – Aktuelle rechtliche, politische, soziale und technisch-organisatorische Fragen. Datenschutz und Datensicherheit – DuD 31(6), 434–440 (2007)
64. Helbach, J., Schwenk, J.: Secure Internet Voting with Code Sheets. In: Alkassar, A., Volkamer, M. (eds.) E-Voting and Identity – First International Conference, VOTE-ID 2007. LNCS, vol. 4896, pp. 166–177. Springer, Heidelberg (2007)
65. Herschberg, M.: Secure Electronic Voting Using the World Wide Web. Master's thesis, MIT (1997)
66. Hirt, M.: Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting. PhD thesis, ETH Zurich (2001)
67. Hirt, M., Sako, K.: Efficient Receipt-Free Voting Based on Homomorphic Encryption. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000)
68. Hof, S.: E-Voting and Biometric Systems. In: Electronic Voting in Europe, pp. 63–72 (2004)
69. Inititative D21 – Arbeitsgruppe 5 'Sicherheit und Vertrauen im Internet': IT-Sicherheitskriterien im Vergleich, http://antareja.rvs.uni--bielefeld.de/˜made/Seminar/eBank/verglaichen-security.pdf.sudah
70. Jakobsson, M.: A Practical Mix. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 448–461. Springer, Heidelberg (1998)
71. Jakobsson, M.: Flash Mixing. In: Proceedings of the 18^{th} ACM Symposium on Principles of Distributed Computing – PODC '99, pp. 83–89. ACM Press, New York (1999)
72. Jakobsson, M., Juels, A., Rivest, R.L.: Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. In: Proceedings of the 11^{th} USENIX Security Symposium, pp. 339–353. USENIX Association, Berkeley (2002)
73. Jefferson, D., Rubin, A.D., Simons, B., Wagner, D.: A Security Analysis of the Secure Electronic Registration and Voting Experiment. SERVE Report (2004), http://www.servesecurityreport.org/
74. Jones, D.W.: Trustworthy Systems on Untrusted Machines. In: Workshop on the Future of Voting Technology in a Networked Environment (2002), http://www.cs.uiowa.edu/~jones/voting/atlanta/
75. Jones, D.W.: Evaluation of voting technology. Advances in Information Security. In: Secure Electronic Voting, pp. 3–16. Kluwer Academic Publishers, Dordrecht (2003)

76. Jonker, H., Volkamer, M.: Compliance of RIES to the Proposed e-Voting Protection Profile. In: Alkassar, A., Volkamer, M. (eds.) E-Voting and Identity – First International Conference, VOTE-ID 2007. LNCS, vol. 4896, pp. 50–61. Springer, Heidelberg (2007)
77. Juels, A., Catalano, D., Jakobsson, M.: Coercion-Resistant Electronic Elections. In: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society – WPES '05, pp. 61–70. ACM Press, New York (2005)
78. Karpen, U.: Elektronische Wahlen? Einige verfassungsrechtliche Fragen. Nomos Verlagsgesellschaft, Baden-Baden (2005)
79. Khorrami, E.: Bundestagswahlen per Internet – Zur rechtlichen und tatsächlichen Realisierbarkeit von Internetwahlen. PhD thesis, Universität Passau (2005)
80. Kim, K., Kim, J., Lee, B., Ahn, G.: Experimental Design of Worldwide Internet Voting System using PKI. In: Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet – SSGRR '01 (2001), http://cris.joongbu.ac.kr/publication/ssgrr2001_kkj.ps
81. Kleinz, T.: IT-Initiative D21 mit erster offiziell gültiger Onlinewahl (2003), http://www.heise.de/newsticker/meldung/42848
82. Kofler, R., Krimmer, R., Prosser, A.: Electronic Voting: Algorithmic and Implementation Issues. In: Proceedings of the 36^{th} Hawaii International Conference on System Sciences – HICSS '03, Information System Track –Organizational Systems and Technology, vol.5, IEEE Computer Society Press, Los Alamitos (2003)
83. Kremer, S., Ryan, M.D.: Analysis of an Electronic Voting Protocol in the Applied Pi-Calculus. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, pp. 186–200. Springer, Heidelberg (2005)
84. Krimmer, R. (ed.): Electronic Voting 2006, 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC. LNI, vol. 86. Gesellschaft für Informatik, Bonn (2006)
85. Krimmer, R., Triessnig, S., Volkamer, M.: The Development of Remote EVoting around the World: A Review of Roads and Directions. In: Alkassar, A., Volkamer, M. (eds.) E-Voting and Identity – First International Conference, VOTE-ID 2007. LNCS, vol. 4896, pp. 1–15. Springer, Heidelberg (2007)
86. Krimmer, R., Volkamer, M.: Bits or Paper? Comparing Remote Electronic Voting to Postal Voting. In: Electronic Government – EGOV '05 – Workshop and Poster Proceedings 4^{th} International EGOV Conference. Schriftenreihe Informatik, vol. 13, pp. 225–232. Universitätsverlag Rudolf Trauner, Linz (2005)
87. Krimmer, R., Volkamer, M.: Wählen auf Distanz: Ein Vergleich zwischen elektronischen und nicht elektronischen Verfahren. In: Effizienz von e-Lösungen in Staat und Gesellschaft, Aktuelle Fragen zur Rechtsinformatik – IRIS '05, pp. 256–262. Boorberg Verlag, Stuttgart (2005)
88. Krimmer, R., Volkamer, M.: Observing Threats to Voter's Anonymity: Election Observation of Electronic Voting. In: Electronic Government – EGOV '06 – Conference Proceedings of the 5^{th} International EGOV Conference. Schriftenreihe Informatik, vol. 18, pp. 43–52. Universitätsverlag Rudolf Trauner, Linz (2006)

89. Krimmer, R., Volkamer, M.: Observing Threats to Voter's Anonymity: Election Observation of Electronic Voting. Working Paper Series on Electronic Participation and Electronic Voting 01/2006, e-Voting.cc (2006), http://www.e-voting.cc/files/Working-Paper-1-2006/
90. Lambrinoudakis, C., Gritzalis, D., Tsoumas, V., Karyda, M., Ikonomopoulos, S.: Secure Electronic Voting: The Current Landscape. In: Secure Electronic Voting. Advances in Information Security, pp. 101–124. Kluwer Academics Publishers, Dordrecht (2003)
91. Lee, B., Kim, K.: Receipt-Free Electronic Voting through Collaboration of Voter and Honest Verifier. In: Proceeding of the Joint Workshop on Information Security and Cryptoloty – JW-ISC '00 (2000), http://caislab.icu.ac.kr/Paper/paper_files/2000/sultan/jwisc2k_bclee.ps
92. Lee, B., Kim, K.: Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 389–406. Springer, Heidelberg (2003)
93. Lipmaa, H.: Electronic Voting Webpage, http://research.cyber.ee/~lipmaa/crypto/link/protocols/voting.php
94. Maaten, E.: Towards Remote E-Voting: Estonian Case. In: Electronic Voting in Europe, pp. 83–100 (2004)
95. Madise, U., Martens, T.: E-Voting in Estonia 2005. The First Practice of Country-wide Binding Internet Voting in the World. In: Krimmer, R. (ed.) Electronic Voting 2006, 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC. LNI, vol. 86, pp. 15–26. Gesellschaft für Informatik, Bonn (2006)
96. Magkos, E., Burmester, M., Chrissikopoulos, V.: Receipt-Freeness in Large-Scale Elections without Untappable Channels. In: Proceedings of the IFIP Conference on Towards The E-Society: E-Commerce, E-Business, E-Government. IFIP Conference Proceedings, vol. 202, pp. 683–693. Kluwer Academics Publishers, Deventer (2001)
97. Mantel, H., Stephan, W., Ullmann, M., Vogt, R.: Guideline for the Development and Evaluation of formal security policy models in the scope of ITSEC and Common Criteria. BSI Document, Version 1.1 (2004)
98. Mauw, S., Verschuren, J., de Vink, E.P.: Data Anonymity in the FOO Voting Scheme. In: Proceedings of the Second International Workshop on Views on Designing Complex Architectures – VODCA '06. ENTCS, vol. 168, pp. 5–28. Elsevier Science Publishers B.V., Amsterdam (2007)
99. McGaley, M.: E-voting: an Immature Technology in a Critical Context. PhD thesis, National University of Ireland (2008)
100. McGaley, M., Gibson, J.P.: A critical analysis of the council of europe recommendations on e-voting. In: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop – EVT '06 (2006), http://www.usenix.org/events/evt06/tech/full_papers/mcgaley/mcgaley.pdf
101. Mercuri, R.: Electronic Vote Tabulation Checks & Balances. PhD thesis, University of Pennsylvania (2000)
102. National Research Council – Committee on a Framework for Understanding Electronic Voting – Computer Science and Telecommunications Board – Devision on Engineering and Physical Sciences: Asking the right questions about Electronic Voting. The National Academies Press, Washington DC (2006)

103. Neff, C.A.: A Verifiable Secret Shuffle and its Applications to E-Voting. In: Proceedings of the 8^{th} ACM Conference on Computer and Communications Security – CCS '01, pp. 116–125. ACM Press, New York (2001)
104. Network Voting System Standards (NVSS). VoteHere Inc. Public Draft 2 (2002)
105. Network Working Group. RFC2119 (1997), http://www.faqs.org/rfcs/rfc2119.html
106. Office for Democratic Institutions and Human Rights: Republic of Estonia – Parliamentary Elections – 4 March 2007 – OSCE/ODIHR Election Assessment Mission Report. Technical report, Organization for Security and Co-operation in Europe (2007)
107. Office for Democratic Institutions and Human Rights: The Netherlands – Parliamentary Elections – 22 November 2006 – OSCE/ODIHR Election Assessment Mission Report. Technical report, Organization for Security and Co-operation in Europe (2007)
108. Office for Democratic Institutions and Human Rights: United States of America – Mid-Term Congressional Elections – 7 November 2006 – OSCE/ODIHR Election Assessment Mission Report. Technical report, Organization for Security and Co-operation in Europe (2007)
109. Office for Democratic Institutions and Human Rights: Swiss Confederation – Federal Elections – 21 October 2007 – OSCE/ODIHR Election Assessment Mission Report. Technical report, Organization for Security and Co-operation in Europe (2008)
110. Office for Democratic Institutions and Human Rights (ODIHR): Election Observation – a decade of monitoring elections: the people and the practise (2005)
111. Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 25–35. Springer, Heidelberg (1998)
112. Ohkubo, M., Miura, F., Abe, M., Fujioka, A., Okamoto, T.: An Improvement on a Practical Secret Voting Scheme. In: Zheng, Y., Mambo, M. (eds.) ISW 1999. LNCS, vol. 1729, pp. 225–234. Springer, Heidelberg (1999)
113. Online-Wahlen Expertengruppe der Gesellschaft für Informatik. GI-Anforderungen an Internetbasierte Vereinswahlen (2005), http://www.gi-ev.de/fileadmin/redaktion/Wahlen/GI-Anforderungen_Vereinswahlen.pdf
114. Oppliger, R., Schwenk, J., Helbach, J.: Protecting Code Voting Against Vote Selling. In: Alkassar, A., Siekmann, J.H. (eds.) Sicherheit. LNI, vol. 128, pp. 193–204. Gesellschaft für Informatik, Bonn (2008)
115. Otten, D.: Mehr Demokratie durch Internetwahlen? Presentation at Nixdorf Forum in Paderborn (2005), http://www.prof-otten.net/uploader/data/MehrDemokratie.pdf
116. Park, C.-s., Itoh, K., Kurosawa, K.: Efficient Anonymous Channel and All/Nothing Election Scheme. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 248–259. Springer, Heidelberg (1994)
117. Physikalisch-Technische Bundesanstalt Braunschweig und Berlin: Prüfbericht – Baumusterprüfung eines Wahlgerätes; PTB-8.51-001.04 (2004), http://www.wahlrecht.de/doku/doku/PB-ESD1-SW03.08-BTW.pdf
118. Pieters, W.: What proof do we prefer? Variants of verifiability in voting. Proceedings of the Workshop on Electronic Voting and e-Government in the U.K. (2006), http://www.cs.ru.nl/~wolterp/Verifiability.pdf

119. Pran, V., Merloe, P.: Monitoring electronic technologies in electoral processes. An NDI Guide for Political Parties and Civic Organizations (2008), http://www.accessdemocracy.org/library/2267_elections_manuals_monitoringtech.pdf
120. Prosser, A., Krimmer, R., Kofler, R.: Implementing an Internet-Based Voting System for Public Elections – A Project Experience. In: Enterprise Information Systems V – ICEIS '03 – 5^{th} International Conference on Enterprise Information Systems, pp. 294–299. Kluwer Academic Publishers, Dordrecht (2004)
121. Puiggali, J.: Applied Cryptography Module C02 – Remote Electronic Voting. Teaching Slides for the esCERT Master on Information Security Technologies (MTSI-5) (2007), http://www.scytl.com/docs/pub/science/Master-ESCERT-07-voting-En.pdf
122. Rat, P.: Grundgesetz für die Bundesrepublik Deutschland (GG) (last change 28.08.2006). Technical report (2006), http://www.datenschutz-berlin.de/recht/de/gg (retrieved on 15-12-2005)
123. Reinhard, K., Jung, W.: Compliance of POLYAS with the BSI Protection Profile – Basic Requirements for Remote Electronic Voting Systems. In: Alkassar, A., Volkamer, M. (eds.) E-Voting and Identity. LNCS, vol. 4896, pp. 62–75. Springer, Heidelberg (2007)
124. Richtlinien für den Einsatz des Digitalen Wahlstift-Systems bei Wahlen zur Hamburgischen Bürgerschaft und Wahlen zu den Bezirksversammlungen sowie bei der Durchführung von Volksentscheiden. Germany (2006), http://fhh.hamburg.de/stadt/Aktuell/pressemeldungen/2006/oktober/31/2006-10-31-bfi-pm-wahl-digitalerstift-wahlgvo-pdf,property=source.pdf (11.06.2008)
125. Rivest, R.L.: The ThreeBallot Voting System (2006), http://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf
126. Rössler, T.: Electronic Voting over the Internet – an E-Government Speciality. PhD thesis, University of Technology Graz (2007)
127. Ryan, P.Y.A., Schneider, S.A.: Prêt à Voter with Re-encryption Mixes. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 313–326. Springer, Heidelberg (2006)
128. Sako, K., Kilian, J.: Secure Voting Using Partially Compatible Homomorphisms. In: Castelfranchi, C., Werner, E. (eds.) MAAMAW 1992. LNCS, vol. 830, pp. 411–423. Springer, Heidelberg (1994)
129. Sako, K., Kilian, J.: Receipt-Free Mix-Type Voting Scheme. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 393–403. Springer, Heidelberg (1995)
130. Saltman, R.G.: Accuracy, Integrity and Security in Computerized Vote-Tallying. Commununication of the ACM 31(10), 1184–1191 (1988)
131. Saltman, R.G.: The History and Politics of Voting Technology – In Quest of Integrity and Public Confidence. Palgrave Macmillan, New York (2006)
132. Scantegrity: The Scantegrity System – An Introductory Whitepaper and Example. Working Draft (2007), http://www.scantegrity.org/papers/whitepaper.pdf
133. Schoenmakers, B.: Fully Auditable Electronic Secret-Ballot Elections. Magazine 1, XOOTIC Alumni Association (2000), http://www.xootic.nl/magazine/jul-2000/schoenmakers.pdf

134. Schulz, U.: Usability-Evaluation. Haw webpage, Hamburg University of Applied Science, Fakultät Design, Medien und Informationen (2006), http://www.bui.haw-hamburg.de/pers/ursula.schulz/webusability/evaluation.html
135. Schweisgut, J.: Coercion-Resistant Electronic Elections with Observer. In: Krimmer, R. (ed.) Electronic Voting 2006, 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC. LNI, vol. 86, pp. 171–177. Gesellschaft für Informatik, Bonn (2006)
136. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
137. Shamos, M.I.: Electronic Voting – Evaluating the Threat. In: Proceedings of the 3rd Conference on Computers, Freedom and Privacy – CPSR '93 (1993), http://www.cpsr.org/prevsite/conferences/cfp93/shamos.html
138. Smith, W.D.: New Cryptographic Election Protocol with Best-known Theoretical Properties. In: Proceedings of the Frontiers in Electronic Elections – FEE '05 (2005), http://www.math.temple.edu/~wds/homepage/jcj.ps
139. The "Regeling voorwaarden en goedkeuring stemmachines". The Netherlands (2006), http://www.wijvertrouwenstemcomputersniet.nl/images/f/fa/Regeling.pdf (last change 2.5.2006)
140. Triessnig, S.: Elektronische Wahlen eine Bestandsaufnahme. Master's thesis, Wirtschaftsuniversität Wien (2007)
141. Ullmann, M., Koob, F., Kelter, H.: Anonyme Online-Wahlen – Lösungsansätze für die Realisierung von Online-Wahlen. Datenschutz und Datensicherheit – DuD (11), 643–647 (2001)
142. U.S. Election Assistance Commission's Technical Guidelines Development Committee (TGDC): Voluntary Voting System Guidelines (VVSG) Recommendations to the Election Assistance Commission (EAC). Draft (2007), http://www.eac.gov/vvsg
143. Verordnung über den Einsatz von Wahlgeräten bei Wahlen zum Deutschen Bundestag und der Abgeordneten des Europäischen Parlaments aus der Bundesrepublik Deutschland: Bundeswahlgeräteverordnung (BWahlGV). Germany (1999), http://bundesrecht.juris.de/bundesrecht/bwahlgv/gesamt.pdf (last change 20.04.1999)
144. Volkamer, M., Alkassar, A., Sadeghi, A.R., Schultz, S.: Enabling the Application of Open Systems like PCs for Online Voting. In: Proceedings of the Frontiers in Electronic Elections – FEE '06 (2006), http://fee.iavoss.org/2006/papers/fee-2006-iavoss-Enabling_the_application_of_open_systems_like-PCs_for_Online_Voting.pdf
145. Volkamer, M., Grimm, R.: Multiple Cast in Online Voting – Analyzing Chances. In: Krimmer, R. (ed.) Electronic Voting 2006, 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC. LNI, vol. 86, pp. 97–106. Gesellschaft für Informatik, Bonn (2006)
146. Volkamer, M., Grimm, R.: Trust Model for Remote Electronic Voting. In: Electronic Government – EGOV '08 – 6th International EGOV Conference – Proceedings of Ongoing Research, Project Contributions and Workshops. Schriftenreihe Informatik, vol. 27, pp. 197–204. Universitätsverlag Rudolf Trauner, Linz (2008)
147. Volkamer, M., Hauff, H.: Zum Nutzen hoher Zertifizierungsstufen nach den Common Criteria (I). Datenschutz und Datensicherheit – DuD (9), 692–695 (2007)

148. Volkamer, M., Hauff, H.: Zum Nutzen hoher Zertifizierungsstufen nach den Common Criteria (II). Datenschutz und Datensicherheit – DuD (10), 766–768 (2007)
149. Volkamer, M., Hutter, D.: From Legal Principles to an Internet Voting System. In: Electronic Voting in Europe, pp. 111–120 (2004)
150. Volkamer, M., Krimmer, R.: Die Online-Wahl auf dem Weg zum Durchbruch. Informatik Spektrum 29(2), 98–113 (2006)
151. Volkamer, M., Krimmer, R.: Overview on Online Voting. In: D*A*CH Security 2006, IT Security & IT Management, pp. 339 –350, Klagenfurt, syssec (2006)
152. Volkamer, M., Krimmer, R.: Secrecy forever? Analysis of Anonymity in Internet-based Voting Protocols. In: Proceedings of the First International Conference on Availability, Reliability and Security – ARES '06, Washington DC, pp. 340–347. IEEE Computer Society Press, Los Alamitos (2006)
153. Volkamer, M., Krimmer, R.: Ver-/Misstrauen Schaffende Maßnahme beim e-Voting. In: Informatik 2006 – Informatik für Menschen, Band 1, Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V (GI). LNI, vol. 93, pp. 418–425. Gesellschaft für Informatik, Bonn (2006)
154. Volkamer, M., Krimmer, R.: Requirements and Evaluation Techniques for Online-Voting. In: Electronic Government – GOV '07 – 6^{th} International GOV Conference – Proceedings of Ongoing Research, Project Contributions and Workshops. Schriftreihe Informatik, vol. 24, pp. 37–46. Universitätsverlag Rudolf Trauner, Linz (2007)
155. Volkamer, M., Krimmer, R., Grimm, R.: Independent Audits of Remote Electronic Voting – Developing a Common Criteria Protection Profile. In: Proceedings of Elektronische Demokratie in Österreich – EDEM '07, pp. 115–126. OCG Verlag, Vienna (2007)
156. Volkamer, M., McGaley, M.: Requirements and Evaluation Procedures for eVoting. In: The Second International Conference on Availability, Reliability and Security – ARES'07, pp. 895–902. IEEE Computer Society Press, Washington DC (2007)
157. Volkamer, M., Reinhard, W., Vogt, R.: Fuse – ein Internetwahlsystem für zeitlich unbegrenzte geheime Betriebsratswahlen. In: Sicherheit 2006: Sicherheit – Schutz und Zuverlässigkeit, Beiträge der 3. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v (GI). LNI, vol. 77, pp. 159–170. Gesellschaft für Informatik, Bonn (2006)
158. Volkamer, M., Vogt, R.: Digitales Wahlstift-System. Common Criteria Protection Profile BSI-PP-0031 (2006), http://www.bsi.de/zertifiz/zert/reporte/PP0031b.pdf
159. Volkamer, M., Vogt, R.: New Generation of Voting Machines in Germany The Hamburg Way to Verify Correctness. In: Proceedings of the Frontiers in Electronic Elections Workshop – FEE '06 (2006), http://fee.iavoss.org/2006/papers/fee-2006-iavoss-New-Generation-of-Voting-Machines-in-Germany.pdf
160. Volkamer, M., Vogt, R.: Basis set of security requirements for Online Voting Products. Commom Criteria Protection Profile BSI-CC-PP-0037 (2008), http://www.bsi.de/zertifiz/zert/reporte/pp0037b_engl.pdf
161. Volkamer, M., Vogt, R.: Basissatz von Sicherheitsanforderungen an Online-Wahlprodukte. Commom Criteria Protection Profile BSI-CC PP-0037 (2008), http://www.bsi.de/zertifiz/zert/reporte/pp0037b.pdf

162. Vollan, K.: Observing Electronic Voting. NORDEM Report 15/2005, The Norwegian Center for Human Rights (2005), http://www.humanrights.uio.no/forskning/publikasjoner/nordem-rapport/2005/1505.pdf
163. VoteHere Webpage: How to do Elections Right? Technical report (03-01-2008), http://www.votehere.net/perspectives/DoVotingRight.pdf
164. VoteHere Webpage: The Metaphysics of Voting. Technical report (03-01-2008), http://www.votehere.net/perspectives/MetaphysicsOfVoting.pdf
165. Voting Equipment Standards: IEEE Project 1583. Technical report (2002), http://grouper.ieee.org/groups/scc38/1583/
166. Verordnung über die politischen Rechte (VPR). Switzerland (2008), http://www.bern.ch/leben_in_bern/stadt/recht/dateien/141.11/Word141.11.pdf (last change 31.03.2008)
167. Weddeling, S., Volkamer, M., Paulsen, C., Mlynczak, K., Meletiadou, A., Meissner, N., Krimmer, R., Helbach, J.: Verifiability in Electronic Voting – An Interdisciplinary View. In: Reduktion der Komplexität durch Recht und IT – IRIS '08 Tagungsband, pp. 165–172. Boorberg Verlag, Stuttgart (2008)
168. Will, M.: Internetwahlen – Verfassungsrechtliche Möglichkeiten und Grenzen. Recht und neue Medien, vol. 2. Richard Boorberger Verlag GmbH & Co, Stuttgart (2002)
169. Williams, B.: Punchcard Voting Systems. SIGSOFT Software Engineering Notes 13(3), 21–21 (1999)
170. Xenakis, A., Macintosh, A.: Procedural security in electronic voting. In: Proceedings of the 37th Hawaii International Conference on System Sciences – HICSS '04, vol. 5, Washington DC, IEEE Computer Society Press, Los Alamitos (2004)

All links have been checked and were working on March 12th 2009. Where they were not working anymore, the last access date has been noted.