Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements

Vom Fachbereich Informatik der

Technischen Universität Darmstadt genehmigte

Dissertation

zur Erlangung des Grades

Doctor rerum naturalium (Dr. rer. nat.)

von

M.Sc. Stephan Rouven Neumann

geboren in Neunkirchen / Saar

Referenten:	Prof. Dr. Melanie Volkamer
	Prof. Dr. Rüdiger Grimm
Tag der Einreichung:	08. Februar 2016
Tag der mündlichen Prüfung:	21. März 2016
Hochschulkennziffer:	D17

Darmstadt 2016

List of Publications

[1] David Bernhard, Stephan Neumann, and Melanie Volkamer. Towards a practical cryptographic voting scheme based on malleable proofs. In *4th International Conference on e-Voting and Identity (VoteID13)*, volume 7985 of *Lecture Notes in Computer Science*, pages 176 – 192. Springer, July 2013.

[2] Johannes Buchmann, Stephan Neumann, and Melanie Volkamer. Tauglichkeit von Common Criteria-Schutzprofilen für Internetwahlen in Deutschland. *Datenschutz und Datensicherheit - DuD*, 38(2):98–102, 2014.

[3] Jurlind Budurushi, Stephan Neumann, Maina Olembo, and Melanie Volkamer. Pretty Understandable Democracy - A Secure and Understandable Internet Voting Scheme. In *8th International Conference on Availability, Reliability and Security (ARES)*, pages 198–207. IEEE, July 2013.

[4] Jurlind Budurushi, Stephan Neumann, Genc Shala, and Melanie Volkamer. Entwicklung eines common criteria schutzprofils für elektronische wahlgeräte mit paper audit trail. In *INF14 - Workshop: Elektronische Wahlen: Unterstützung der Wahlprozesse mittels Technik*, volume 232 of *Lecture Notes in Informatics (LNI)*, pages 1415–1426. Gesellschaft für Informatik e.V. (GI), Köllen Druck+Verlag GmbH, Bonn, September 2014.

[5] Jurlind Budurushi, Stephan Neumann, and Melanie Volkamer. Smart cards in electronic voting - lessons learned from applications in legally binding elections and approaches proposed in scientific papers. In *5th International Conference on Electronic Voting (EVOTE)*, volume 205 of *LNI - Series of the Gesellschaft für Informatik (GI)*, pages 257–270. Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC, Gesellschaft für Informatik, July 2012.

[6] Christian Feier, Stephan Neumann, and Melanie Volkamer. Coercion-Resistant Internet Voting in Practice. In *INF14 - Workshop: Elektronische Wahlen: Unterstützung der Wahlprozesse mittels Technik*, volume 232 of *Lecture Notes in Informatics (LNI)*, pages 1401–1414. Gesellschaft für Informatik e.V. (GI), Köllen Druck+Verlag GmbH, Bonn, September 2014.

[7] Oksana Kulyk, Karola Marky, Stephan Neumann, and Melanie Volkamer. Introducing proxy voting to helios. In *11th International Conference on Availability, Reliability and Security (ARES)*, September 2016.

[8] Oksana Kulyk, Stephan Neumann, Jurlind Budurushi, Melanie Volkamer, Rolf Haenni, Reto Koenig, and Philemon von Bergen. Efficiency evaluation of cryptographic protocols for boardroom voting. In *10th International Conference on Availability, Reliability and Security (ARES)*, pages 224–229. IEEE, August 2015.

[9] Oksana Kulyk, Stephan Neumann, Karola Marky, Jurlind Budurushi, and Melanie Volkamer. Coercion-resistant proxy voting. In *ICT Systems Security and Privacy Protection - 31st IFIP TC 11 International Conference, SEC 2016 Proceedings*, pages 3–16, 2016.

[10] Oksana Kulyk, Stephan Neumann, Melanie Volkamer, Christian Feier, and Thorben Köster. Electronic voting with fully distributed trust and maximized flexibility regarding ballot design. In *6th International Conference on Electronic Voting (EVOTE)*, pages 1–10. IEEE, 2014.

[11] Peter Meyer, Stephan Neumann, and Melanie Volkamer. Supporting decision makers in choosing suitable authentication schemes. In *International Symposium on Human Aspects of Information Security & Assurance (HAISA 2016)*, 2016. Accepted for publication.

[12] Stephan Neumann, Jurlind Budurushi, and Melanie Volkamer. *Analysis of Security and Cryptographic Approaches to Provide Secret and Verifiable Electronic Voting*, chapter 2, pages 27–61. Design, Development, and Use of Secure Electronic Voting Systems. IGI Global, 2014.

[13] Stephan Neumann, Christian Feier, Perihan Sahin, and Sebastian Fach. Pretty Understandable Democracy 2.0. In *6th International Conference on Electronic Voting (EVOTE)*, pages 69 – 72. TUT Press, October 2014.

[14] Stephan Neumann, Christian Feier, Melanie Volkamer, and Reto Koenig. Towards A Practical JCJ / Civitas Implementation. In *INF13 - Workshop: Elektronische Wahlen: Ich sehe was, das Du nicht siehst - öffentliche und geheime Wahl*, pages 804–818, July 2013. A revised version of this work is available under http://eprint.iacr.org/2013/464.

[15] Stephan Neumann, Anna Kahlert, Maria Henning, Hugo Jonker, and Melanie Volkamer. Informatische Modellierung der Prinzipien des gesetzlichen Gestaltungsspielraums im Hinblick auf Wahlsysteme. In *Abstraction and Application: Proceedings of the 16th International Legal Informatics Symposium (IRIS)*, pages 277–284, February 2013.

[16] Stephan Neumann, Anna Kahlert, Maria Henning, Philipp Richter, Hugo Jonker, and Melanie Volkamer. Modeling the German Legal Latitude Principles. In *5th International Conference on eParticipation (ePart)*, volume 8075 of *Lecture Notes in Computer Science*, pages 49–56. Springer, September 2013.

[17] Stephan Neumann, Oksana Kulyk, and Melanie Volkamer. A usable android application implementing distributed cryptography for election authorities. In *9th International Conference on Availability, Reliability and Security (ARES)*, pages 207–216. IEEE, September 2014.

[18] Stephan Neumann, Maina M. Olembo, Karen Renaud, and Melanie Volkamer. Helios verification: To alleviate, or to nominate: Is that the question, or shall we have both? In *3rd International Conference on Electronic Government and the Information Systems Perspective*, volume 8650 of *Lecture Notes in Computer Science*, pages 246–260. Springer, September 2014.

[19] Stephan Neumann and Melanie Volkamer. Civitas and the Real World: Problems and Solutions from a Practical Point of View. In *7th International Conference on Availability, Reliability and Security (ARES)*, pages 180–185. IEEE, August 2012.

[20] Stephan Neumann and Melanie Volkamer. Formal treatment of distributed trust in electronic voting. volume 7th International Conference on Internet Monitoring and Protection, pages 30–39. IARIA, May 2012.

[21] Stephan Neumann and Melanie Volkamer. *A Holistic Framework for the Evaluation of Internet Voting Systems*, chapter 4, pages 76–91. Design, Development, and Use of Secure Electronic Voting Systems. IGI Global, 2014.

[22] Stephan Neumann, Melanie Volkamer, Jurlind Budurushi, and Marco Prandini. SecIVo: A quantitative security evaluation framework for internet voting schemes. *Annals of Telecommunications*, pages 1–16, 2016.

[23] Stephan Neumann, Melanie Volkamer, Moritz Strube, Wolfgang Jung, and Achim Brelle. Cast-as-intended-verifizierbarkeit für das polyas-internetwahlsystem. *Datenschutz und Datensicherheit*, 39(11):747–752, 2015.

[24] Maina M. Olembo, Anna Kahlert, Stephan Neumann, and Melanie Volkamer. Partial Verifiability in POLYAS for the GI Elections. In *5th International Conference on Electronic Voting (EVOTE)*, volume 205 of *LNI - Lecture Notes in Informatics*, pages 95–109. Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC, Gesellschaft für Informatik, July 2012.

[25] Alexander Roßnagel, Philipp Richter, Anna Kahlert, Melanie Volkamer, Stephan Neumann, Rüdiger Grimm, and Daniela Simić-Draws. Holistic and Law compatible IT

Security Evaluation: Integration of Common Criteria, ISO 27001/IT-Grundschutz and KORA. *International Journal of Information Security and Privacy (IJISP)*, 7(3):16–35, 2013.

[26] Fatemeh Shirazi, Stephan Neumann, Ines Ciolacu, and Melanie Volkamer. Robust Electronic Voting: Introducing Robustness in Civitas. In *International Workshop on Requirements Engineering for Electronic Voting Systems (REVOTE)*, pages 47 –55, August 2011.

[27] Peter Wolf, Jordi Barrat, Eden Bolo, Alejandro Bravo, Robert Krimmer, Stephan Neumann, Al A. Parreño, Carsten Schürmann, and Melanie Volkamer. *Certification of ICTs in Elections*. 2015.

List of Figures

Acknowledgments

Der Herr ist mein Hirte, mir wird nichts mangeln. Er weidet mich auf einer grünen Aue und führet mich zum frischen Wasser. Er erquicket meine Seele. Er führet mich auf rechter Straße um seines Namens willen. Und ob ich schon wanderte im finstern Tal, fürchte ich kein Unglück; denn du bist bei mir, dein Stecken und Stab trösten mich. Du bereitest vor mir einen Tisch im Angesicht meiner Feinde. Du salbest mein Haupt mit Öl und schenkest mir voll ein. Gutes und Barmherzigkeit werden mir folgen mein Leben lang, und ich werde bleiben im Hause des Herrn immerdar.

Psalm 23

Der größte Dank gilt meiner Betreuerin, Prof. Dr. Melanie Volkamer. Ich danke ihr dafür, dass sie mir die große Chance gegeben hat, an der Technischen Universität Darmstadt zu promovieren und sie mir ihr Vertrauen und ihre Unterstützung in jeder Phase meiner Promotion geschenkt hat. Unter ihrer Leitung konnte ich mich sowohl wissenschaftlich wie auch persönlich permanent weiterentwickeln. Vielen Dank für alles, Melanie!

Ich danke Prof. Dr. Rüdiger Grimm, der mir im Laufe meiner Promotion immer wieder mit Rat und Tat zur Seite stand und schließlich auch die Ko-Betreuung übernahm.

Ein großer Dank geht an Prof. Dr. Buchmann, dessen Unterstützung mir den Weg an die Technische Universität Darmstadt bereitet hat.

Ich bin dankbar und geehrt, Prof. Dr. Reiner Hähnle, Prof. Dr. Alejandro Buchmann und Prof. Dr. Kay Hamacher als Mitglieder meiner Prüfungskommission zu haben.

Ein großes Dankeschön gilt meinen Kollegen der Forschungsgruppen SECUSO und CDC. Ich danke insbesondere meinen Kollegen Jurlind Budurushi und Oksana Kulyk. Ohne ihre sowohl kritischen Fragen und Hinweise, wie auch konstruktiven und aufbauenden Ideen und Anregungen sowie die gefühlt unendliche Zeit, die sie für mich aufgebracht haben, wäre es vielleicht nie zum Abschluss dieser Dissertation gekommen. Ich danke Euch!

Ein großer Dank gilt all meinen Mitautoren.

Ich danke Prof. Dr. Rolf Haenni, Prof. Dr. Reto Koenig und Prof. Dr. Marco Prandini, deren Arbeitsgruppen ich während meiner Promotion besuchen durfte.

Ich denke Joshua Ruf, der intensiv an den Implementierungen dieser Arbeit mitgewirkt hat.

Ich danke Manuel Noll, der mir mit seiner überaus großen Hilfsbereitschaft und seinem mathematischen Wissen zu jeder Tages- und Nachtzeit zur Seite stand. Ich bedanke mich bei Dr. David Bernhard, der mir ebenfalls oftmals helfend zur Seite stand. Ich danke Dr. Steffen Bartsch, der mir als Kollege und Freund immer wieder wertvolle Denkanstöße gab.

Ich danke Lassaad Cheikhrouhou und PD Dr. Werner Stephan, die mich bei meinen ersten Schritten im wissenschaftlichen Umfeld intensiv und nachhaltig unterstützt haben.

Ich bedanke mich für die finanzielle Unterstützung durch die Horst Görtz Stiftung.

Ich danke meinen Eltern dafür, dass sie mir das Geschenk des Lebens gemacht haben, dass sie mir die Möglichkeit gegeben haben, diesen Karriereweg einzuschlagen und dass sie in jeder Situation meines Lebens rückhaltlos hinter mir stehen. Ihnen sei diese Arbeit gewidmet.

Ich danke meiner Freundin Charlene Threm, die in den letzten Jahren jederzeit und unermüdlich für mich da war und mir jeden Tag die Kraft gegeben hat, diese Aufgabe zu meistern.

Ich danke meiner “Saarbrücker Verbindung”, Pavlo Lutsik, Ayman Haidar, Hassan El Soumsoumani und ihren Familien.

Ich danke all meinen Freunden und Verwandten, die mich zu dem werden ließen, was ich heute bin.

Abstract

In recent years, several nations and private associations have introduced Internet voting as additional means to conduct elections. To date, a variety of voting schemes to conduct Internet-based elections have been constructed, both from the scientific community and industry. Because of its fundamental importance to democratic societies, Internet voting – as any other voting method – is bound to high legal standards, particularly imposing security requirements on the voting method. However, these legal standards, and resultant derived security requirements, partially oppose each other. As a consequence, Internet voting schemes cannot enforce these legally-founded security requirements to their full extent, but rather build upon specific assumptions. The criticality of these assumptions depends on the target election setting, particularly the adversary expected within that setting. Given the lack of an election-specific evaluation framework for these assumptions, or more generally Internet voting schemes, the adequacy of Internet voting schemes for specific elections cannot readily be determined. Hence, selecting the Internet voting scheme that satisfies legally-founded security requirements within a specific election setting in the most appropriate manner, is a challenging task.

To support election officials in the selection process, the first goal of this dissertation is the construction of a evaluation framework for Internet voting schemes based on legally-founded security requirements. Therefore, on the foundation of previous interdisciplinary research, legally-founded security requirements for Internet voting schemes are derived. To provide election officials with improved decision alternatives, the second goal of this dissertation is the improvement of two established Internet voting schemes with regard to legally-founded security requirements, namely the Polyas Internet voting scheme and the Estonian Internet voting scheme.

Our research results in five (partially opposing) security requirements for Internet voting schemes. On the basis of these security requirements, we construct a capability-based risk assessment approach for the security evaluation of Internet voting schemes in specific election settings. The evaluation of the Polyas scheme reveals the fact that compromised voting devices can alter votes undetectably. Considering surrounding circumstances, we eliminate this shortcoming by incorporating out of band codes to acknowledge voters' votes. It turns out that in the Estonian scheme, four out of five security requirements rely on the correct behaviour of voting devices. We improve the Estonian scheme in that regard

by incorporating out of band voting and acknowledgment codes. Thereby, we maintain four out of five security requirements against adversaries capable of compromising voting devices.

Zusammenfassung

In den letzten Jahren ist ein allgemeiner Trend in Richtung Internetwahlen zu beobachten. So hat sich die Stimmabgabe über das Internet als zusätzlicher Wahlkanal in einigen Staaten und privaten Vereinigungen etabliert. Bis zum heutigen Tag haben sowohl Wissenschaft wie auch Industrie eine Reihe von Internetwahlprotokollen zur Durchführung von Internetwahlen entwickelt. Aufgrund ihrer zentralen Bedeutung für demokratische Gesellschaften ist die Internetwahl – wie jede andere Wahlmethode – an hohe rechtliche Normen gebunden. Insbesondere erlegen diese Normen der Internetwahl Sicherheitsanforderungen auf. Es zeigt sich jedoch, dass die rechtlichen Normen sowie die daraus abgeleiteten Sicherheitsanforderungen miteinander konkurrieren. Eine Konsequenz dieser Tatsache ist, dass Internetwahlprotokolle die rechtlich begründeten Sicherheitsanforderungen nur unter bestimmten Annahmen umsetzen können. Die Kritikalität dieser Annahmen hängt dabei von der Wahlumgebung ab, insbesondere von dem zu erwartenden Angreifer. Aufgrund des Fehlens wahlabhängiger Evaluationsmethoden für diese Annahmen, oder genereller für Internetwahlprotokolle, können verschiedene Internetwahlprotokolle nicht direkt auf ihre Eignung zum Einsatz zur Durchführung bestimmter Wahlen untersucht werden. Folglich fällt Wahlverantwortlichen die Auswahl eines Internetwahlprotokolls, das die rechtlich begründeten Sicherheitsanforderungen in einer gegebenen Wahlumgebung bestmöglich umsetzt, schwer.

Um Wahlverantwortliche bei dieser Auswahl zu unterstützen, definieren wir die Konstruktion einer Evaluationsmethode für Internetwahlprotokolle bezüglich rechtlich begründeter Sicherheitsanforderungen als erstes Ziel dieser Dissertation. Dazu werden auf Grundlage interdisziplinärer Vorarbeit rechtlich begründete Sicherheitsanforderungen für Internetwahlprotokolle abgeleitet. Um Wahlverantwortliche darüber hinaus mit gegebenenfalls besseren Alternativen zu unterstützen, ist das zweite Ziel dieser Dissertation die Verbesserung etablierter Internetwahlprotokolle bezüglich rechtlich begründeter Sicherheitsanforderungen. Dazu werden das Polyas Internetwahlprotokoll sowie das Protokoll des estnischen Internetwahlsystems betrachtet.

Die Forschungsergebnisse dieser Dissertation resultieren in fünf (teilweise konkurrierenden) Sicherheitsanforderungen für Internetwahlprotokolle. Auf Grundlage dieser Anforderungen konstruieren wir einen fähigkeitsbasierten Ansatz zur Risikoabschätzung zur Sicherheitsevaluation von Internetwahlprotokollen in bestimmten Wahlumgebungen. Die

Evaluation des Polyas Protokolls legt die Tatsache offen, dass kompromittierte Wahl-Endgeräte abgegebene Stimmen unbemerkt manipulieren können. In Einklang mit praktischen Gegebenheiten adressieren wir die Schwachstelle durch das Einarbeiten sogenannter Bestätigungscodes. Die Evaluation des estnischen Internetwahlprotokolls zeigt, dass vier der fünf rechtlich begründeten Sicherheitsanforderungen nur unter der Annahme gewährleistet werden können, dass Wahl-Endgeräte nicht kompromittiert sind. Wir begegnen dieser Schwachstelle mit der Einarbeitung sogenannter Wahl- und Bestätigungscodes. Das Ergebnis dieser Erweiterung ist, dass vier der fünf rechtlichen begründeten Sicherheitsanforderungen nicht durch kompromittierte Endgeräte gefährdet werden.

Contents

Chapter 1

Introduction

The first chapter provides the reader an introduction into the content of this thesis. We motivate our research and define the research questions addressed within the thesis. We subsequently provide an overview about the contributions of the thesis. Finally, we guide the reader through the remainder of this work.

1.1. Motivation and Research Questions

The history of election dates back to the ancient Greece (508/07 BC) when citizens of Athens for the first time exercised a direct democracy. However, at that time the electoral right was very limited. Only citizens of Athens excluding women, metics, and slaves were allowed to participate in the elections. Today, the role of election for democratic societies is nearly beyond any doubt. In fact, the right to political participation is anchored in Universal Declaration of Human Rights [Uni48]. Article 21 of this declaration states:

1. *"Everyone has the right to take part in the government of his country, directly or through freely chosen representatives."*

2. *"Everyone has the right of equal access to public service in his country."*

3. *"The will of the people shall be the basis of the authority of government; this will shall be expressed in periodic and genuine elections which shall be by universal and equal suffrage and shall be held by secret vote or by equivalent free voting procedures."*

While the Human Rights Declaration captures the three *election principles* universal, equal, and free suffrage, other nations extend these principles. Throughout this work, we focus on Germany. As baseline for our research, we consider Federal elections in Germany. According to the German Constitution, the principles of the universal, direct, free, equal, and secret elections established in Art. 38.1 sentence 1 of the German Constitution are of particular relevance. In addition, the principle of the public nature of elections emerges from Articles 20.1, 20.2 and 38.1 of the German Constitution.

The conduct of elections has always been under change, incorporating technical advances of human development to facilitate or generally improve the voting process. While in the ancient Greece, citizens deposited shreds or pebbles into distinct accumulation bins to express their choice, citizens in the ancient Rome used paper ballots to indicate the name of their preferred candidate. In the year 1856, the Australian government for the first time issued uniform ballots to their citizens upon which they expressed their choice. The advance of the industrialization resulted in the next stage of development. In 1892, for the first time mechanical voting machines (lever voting machines) were used to conduct elections in Lockport, New York, USA. With digitalization, the most recent stage of development has been initiated. Direct-recording electronic (DRE) voting machines have been in use since the early 1990s [FC05]. Lately, a tendency towards the usage of Internet-based voting systems can be noticed. Estonia has taken a leading role with the invention of Internet voting for local government council elections and nation-wide parliamentary elections in 2005 [Kal09]. Following Estonia's example, other countries started adopting Internet voting for legally-binding elections, *e.g.* Canada, Switzerland, India, and Norway[1].

While new (compulsory or optional) voting modes generally address deficiencies with regard to legal provisions in previous voting modes, those provisions cannot be satisfied simultaneously in totality [Fed, Decision: 59, 119 (124):1981]. In its judgment on the constitutionality of postal voting, the Federal Constitutional Court declared that the principles of the free and secret elections were not violated by the postal voting process [Fed, Decision: 21, 200:1967]: the increase in election participation offered by postal voting, which translates to an improvement of the principle of the universal elections, is strong enough to offset the impairment of the secret elections, and thus can be accepted. Before introducing Internet-based elections, election officials therefore have to gain a profound understanding to which extent Internet voting systems satisfy legal provisions.

In the context of Internet voting, legal provisions have largely been considered in a constructive approach, *i.e.* researchers strove to deduce precise technical goals and viable assumptions, and tailor their constructions towards these goals under the given assumptions. Among other methodologies, *e.g.* [VH04, BGRR13], a well-established one following this approach is the *Common Criteria for Information Technology Security Evaluation*[2] (DIN ISO/IEC 15408). Intuitively, these approaches might either result in technical constructions that do or do not comply with technical goals under the given assumptions. Because election principles cannot be enforced to their full extent within Internet voting systems, legal latitude is open for the legislator [Dre06, Art. 38, Rn. 62]. Hence, there is no unique set of technical goals and assumptions. Therefore, it is reasonable to evaluate Internet voting systems against legally-founded technical requirements, taking the election environment into account, including the expected adversary. The present thesis supports

[1] Refer to `http://aceproject.org/ace-en/focus/e-voting/countries`
[2] Refer to `https://www.commoncriteriaportal.org/cc/`

this evaluation approach. Its focus is thereby restricted to Internet voting schemes as conceptual underpinning of Internet voting systems and to security aspects of these schemes. Consequently, we pose our first research question:

Research Question 1. *How can the satisfaction of legally-founded security requirements in Internet voting schemes be measured?*

To date, numerous Internet voting schemes have been proposed [CGS97, JCJ05, RT13, ZCC+13] and applied for different types of elections, *e.g.* Estonian parliamentary elections [Tre07, Off11, HLV12], Norwegian parliamentary elections [OSC12], University elections [ADMPQ09], and elections in private associations [OKNV12]. The second research question tackles the challenge of evaluating the security of established Internet voting schemes and potentially improving the security of these schemes. Hence, the following research question is defined:

Research Question 2. *Can established Internet voting schemes be improved with regard to legally-founded security requirements for Internet voting schemes?*

1.2. Research Approaches and Contributions

The thesis contributes to the advancement of Internet voting by addressing the defined research questions. An overview about the respective research approaches and contributions is provided in the following paragraphs.

Evaluation of Internet Voting Schemes Based on Legally-Founded Security Requirements

The implementation of Internet-based elections is bound to legal provisions, most generally expressed in the election principles. Because of their abstract nature, the evaluation of Internet voting systems against these provisions requires a refinement of legal provisions into technical requirements.

As a first contribution of this thesis, we pave the way for an evaluation of Internet voting systems with regard to legal provisions by transforming election principles and further constitutional rights relevant to Internet voting into *security requirements* for Internet voting systems. Therefore, on the foundation of previous research conducted by Bräunlich *et al.* [BGRR13], we transform legal criteria/technical requirements derived from legal provisions on security requirements for Internet voting systems.

To measure the extent to which Internet voting schemes as conceptual underpinning of Internet voting systems satisfy security requirements, as a second contribution of this thesis, we construct a *security evaluation framework* for Internet voting schemes. The framework provides two specification languages on the basis of uniform adversarial capabilities. The language of qualitative security models enables system analysts to specify the

security of Internet voting schemes in an election-independent manner. To that end, qualitative security models serve as first quality criterion for Internet voting schemes, upon which a notion of *dominance* can be defined. The language of election settings allows election officials to specify their election environment in terms of expected adversaries, number of eligible voters, and number of expected voters. Ultimately, the framework allows the evaluation of given qualitative security models within a given election setting by the application of a risk-based approach and Monte-Carlo simulations. The quantitative results serve as second quality criterion for Internet voting schemes.

Security Evaluation and Improvement of Internet Voting Schemes

Because of the lack of appropriate evaluation techniques, the security of Internet voting schemes and their modifications have not been evaluated against legally-founded security requirements. We therefore present security evaluations of well-established Internet voting schemes, namely the Polyas Internet voting scheme and the Estonian Internet voting scheme.

As a third contribution of this thesis, we qualitatively evaluate the Polyas Internet voting scheme against the uniform capability set. On the basis of the qualitative evaluation result, we incorporate a *verifiability measure* into the Polyas Internet voting scheme to uphold vote integrity against compromised voting devices. We compare the original and the extended Polyas scheme both on a qualitative and quantitative level.

Analogously to the Polyas case, as a fourth contribution of this thesis, we qualitatively evaluate the Estonian Internet voting scheme against the uniform capability set. We subsequently construct an *extension* of the Estonian Internet voting scheme, thereby improving the scheme with regard to risks caused by compromised voting devices. We compare the original and the extended Estonian scheme both on a qualitative and quantitative level.

1.3. Structure and Preliminary Considerations

We provide the structure of this thesis and outline preliminary considerations relevant for the remainder of this work.

Structure. Aligned with its research questions, the content of this thesis is subdivided in two parts. An overview about the thesis structure is provided in Figure 1.1.

Part I of this thesis is dedicated to the construction of an evaluation framework for Internet voting schemes based on legally-founded security requirements. Therefore, in Chapter 2, we derive security requirements for Internet voting systems on the basis of election principles manifested in the German Constitution and further relevant constitutional rights. The actual construction of a security evaluation framework for Internet voting schemes, as conceptual underpinning of Internet voting systems, is presented in Chapter 3.

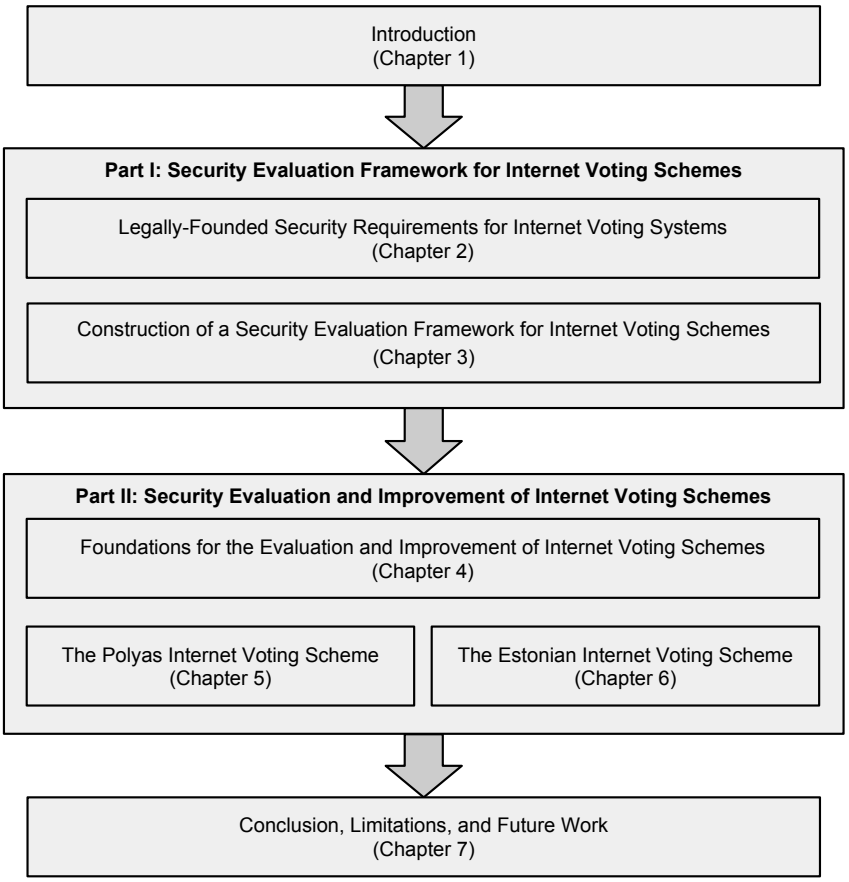

Figure 1.1: Structure of this dissertation thesis.

Part II of this thesis is dedicated to the quantitative security evaluation of established Internet voting schemes and their potential improvement. Chapter 4 provides the reader with the foundations for the evaluation and improvement of Internet voting schemes. These are cryptographic primitives and protocol upon which Internet voting schemes build and probabilistic adversaries which are used throughout the quantitative security evaluation of Internet voting schemes. Subsequently, the Polyas Internet voting scheme is evaluated and improved with regard to vote integrity in Chapter 5. Analogously, we evaluate the Estonian Internet voting scheme and propose an extension in Chapter 6.

The conclusions, limitations, and directions for future research of the thesis are discussed in Chapter 7.

Preliminary Considerations. The goal of this work is the evaluation and improvement of Internet voting schemes with regard to legally-founded security requirements. To achieve this goal, it is necessary to isolate these schemes from possibly alternative voting methods, such as postal voting and polling station voting. To draw this line precisely from the beginning, we consider Internet voting as compulsory voting method in the remainder of this work.

We disassemble Internet voting into three phases. Within the first phase, namely the *setup phase*, election specific data is generated and distributed. In that phase, all provisions for the actual voting process are made. Within the second phase, namely the *voting phase*, voters have the possibility to actively participate in the election by casting their vote. Within the third phase, namely the *tallying phase*, all votes cast throughout the voting phase are tallied and the election result is announced. Depending on the scheme, further specific election data is published.

All links provided within this thesis have been checked and were working on February 9, 2016.

Part I.

Security Evaluation Framework for Internet Voting Schemes

Chapter 2

Legally-Founded Security Requirements for Internet Voting Systems

The goal of this chapter is the derivation of security requirements for Internet voting systems on the basis of election principles anchored in the German Constitution and further constitutional rights relevant to Internet voting.

We first review scientific works dedicated to the derivation of security requirements for Internet voting systems from legal provisions. We conclude that none of these works provides a satisfactory list of security requirements as foundation for this work. We subsequently build upon the interdisciplinary research by Bräunlich *et al.* [BGRR13] on deriving technical design proposals from legal provisions anchored in the German Constitution. We derive security requirements for Internet voting systems in the third part of this chapter. We summarize the content of this chapter in the last section.

An earlier version of this chapter has been published as chapter in the book *Design, Development, and Use of Secure Electronic Voting Systems* [21].

2.1. Related Work

Throughout the last decades, many researchers have addressed the challenge of establishing security requirements for Internet voting systems on the basis of legal provisions. This section reviews those works and draws the line between efforts made earlier and our own contribution.

Gritzalis [Gri02] aims at bridging the gap between legal provisions and technical requirements. Therefore, Gritzalis first identifies a set of constitutional requirements, namely generality, freedom, equality, secrecy, directness, and democracy. Subsequently, the author derives voting system design principles. Applying the Rational Unified Process [JBR+99, Kru04], the author refines the constitutional requirements and respective design principles into twelve user requirements. Mitrou *et al.* [MGK02] address the question *"how an e-vote process should be designed and implemented in order to comply with the democratic election principles"*. The authors focus on the election principles of univer-

sal, free, equal, secret, and direct voting; additionally, they emphasize the importance of transparency, verifiability, accountability, security and accuracy. Both works are settled in an international context, rather than the German context. Given the fact that legal interpretations of election principles might differ in different contexts (refer for instance to Grimm *et al.* [GKM+06]), the goal of our work is the derivation of security requirements for Internet voting systems in the German context.

A number of scientific works have studied the impact of legal provisions stemming from the German Constitution on the implementation of electronic voting.

Schryen [Sch04] studies the relation between (technological) security and surrounding requirements such as legal, economical, ergonomic, and other requirements. As a baseline of a legal surrounding, Schryen builds upon the election principles of the German Constitution. The author elaborates concerns that arise with regard to election principles when introducing electronic voting, and shows how these concerns might be addressed by technological means. In spite of these insights, the author does not structurally derive security requirements from legal provisions for Internet voting systems.

Volkamer and Hutter [VH04] provide a technical interpretation of these principles and investigate how an electronic voting system could be implemented to accommodate these requirements. Aiming at a general catalogue of security requirements for elections at the Gesellschaft für Informatik, Grimm *et al.* [GKM+06] interpret the legal provisions of the German Constitution and deduce nine security requirements for Internet voting systems. These efforts and numerous further works build the basis of Volkamer's PhD thesis [Vol09]. The goal of her thesis is the establishment of a comprehensive set of system requirements (including security requirements) for electronic voting systems. Therefore, Volkamer reviews the Federal Regulation for Voting Machines of the Federal Republic of Germany (*Verordnung über den Einsatz von Wahlgeräten bei Wahlen zum Deutschen Bundestag und der Abgeordneten des Europäischen Parlaments aus der Bundesrepublik Deutschland*), the guidelines about the usage of the digital voting pen system of the Free and Hanseatic City of Hamburg (*Richtlinien für den Einsatz des Digitalen Wahlstift-Systems bei Wahlen zur Hamburgischen Bürgerschaft und Wahlen zu den Bezirksversammlungen*), the Voting System Standard of the Federal Election Commission of the United States of America [Fed01], Voluntary Voting System Guidelines [Com07], the IEEE P1583 Standards for Voting Equipment, the CoE Recommendation Rec(2004)11 [Cou04], the Catalogue of Requirements for Online Voting Systems for Non-parliamentary Elections [Phy04] by the Physikalisch-Technische Bundesanstalt (PTB), the requirements catalogue for Internet-based election in private associations [Ges05] by the Gesellschaft für Informatik, the Swiss Election Law, the Austrian Federal Law about the representation of the students (*Bundesgesetz über die Vertretung der Studierenden (Hochschülerinnen- und Hochschülerschaftsgesetz 1998 - HSG 1998)*), the Network Voting System Standards (NVSS) [Vot02]. Additionally, Volkamer considers scientific works aiming at the derivation of technical requirements for electronic voting systems, namely Shamos Commandments [Sha93], Mercuri's

PhD thesis [Mer01], McGaley's PhD thesis [McG08] and the voting system requirements in the CyberVote project [FTLB01]. Her research results in 21 security requirements for Internet voting systems. In spite of the enormous ambition to include both legal provisions and technically-driven requirements, the requirements derived by Volkamer prove to be inadequate for this work. The derived security requirements show redundancy and are too closely related to technical solutions. Consider for instance the requirements *O.T.ElecSecrecyNet*, *O.T.ProofGen*, and *O.T.ElectionSecrecy*. The purpose of these three requirements is the same, namely preventing adversaries from learning the link between a voter and her vote. Consider furthermore the requirement *O.T.IneligVoter*. It requires voting systems to *"unambiguously identify and authenticate the voter before storing his vote in the e-ballot box"*. Yet, there exist approaches which enforce the equality of voters by means of anonymous credentials, *e.g.* linkable group signatures [LWW04] and distributed credentials [JCJ05].

Given the lack of an adequate list of security requirements structurally derived from legal provisions captured in the German Constitution, we address this challenge in the following sections.

2.2. Preliminary Work – Refinement of Constitutional Rights

To the best of our knowledge, Bräunlich *et al.* [BGRR13] present the first interdisciplinary collaboration structurally refining election principles and further relevant rights anchored in the German Constitution into technical design design proposals. We first present the election principles anchored in the German Constitution and further constitutional rights related to the Internet voting process. Subsequently, we outline the refinement process and result of the interdisciplinary research conducted by Bräunlich *et al.* We identify shortcomings of their approach in reference to our research goal. For the sake of clarity, we cite excerpts of their work. Those excerpts have been translated from German to English and printed in *italic font* between quotation marks.

2.2.1. Election Principles and Further Constitutional Rights

The election of the representatives is regulated in Article 38 of the German Constitution. Correspondingly, the principles of the *universal*, *direct*, *free*, *equal*, and *secret* elections established in Article 38.1 sentence 1 are of particular relevance. In addition to these principles, another election principle emerging from Article 20.1, 20.2 and 38.1 of the German Constitution has been emphasized by the Federal Constitutional Court in 2009 [Fed, Decision: 123, 39:2009], namely the principle of the *public nature* of elections.

Universal Elections. The principle of universal elections concerns the eligibility to vote without applying to personal qualities or political, financial or social aspects [Fed, Decision:

15, 165 (166f):1962. Decision: 36, 139 (141):1973].

Equal Elections. The principle of equal elections addresses the impact of every valid vote on the election result. That is, every voter needs to have the same number of votes and must be able to cast his or her vote in the same way as any other one [Sch09, § 1, Rn. 43]. Furthermore, all candidates need to be presented equally, so all of them have the same chance to win the election [Sch09, § 1, Rn. 48f].

Direct Elections. The principle of direct elections forbids the integration of electoral delegates [Fed, Decision: 7, 63 (68):1957. Decision: 47, 253 (279):1978] and requires that the representatives get elected through voters only by casting their vote personally [Dre06, Art. 38, Rn. 75][MD13, Art. 38, Rn. 101].

Secret Elections. The principle of secret elections claims that the voting decision remains secret during and after the election process [vMK12, Art. 38, Rn. 67]. It needs to remain secret whether voters split their votes or cast them based on a single preferred party, whether they spoiled their vote or abstained from voting at all [Sch09, § 1, Rn. 95].

Free Elections. The principle of free elections covers the process of opinion making prior to the election as well as the process of vote casting within the election. In formal aspects it ensures the right to choose whether one wants to casts a vote or not. In material regards it provides the freedom to cast a vote for the preferred candidate or party [Sch09, § 1, Rn. 21].

Public Nature of Elections. The so called *public nature* of elections requires that all essential steps in the elections are subject to public examinability unless other constitutional interests justify an exception.

In addition to the election principles anchored in the German Constitution, Bräunlich *et al.* identify two further constitutional rights relevant to the implementation of Internet voting. These are the following:

Informational Self-determination. The informational self-determination goes back to the "Census Judgment" by the German Constitutional Court [Fed, Decision: 65, 1:1983] and is deduced by the Art. 1(1) and Art. 2(1) of the German Constitution. In that judgment, the informational self-determination is established as Basic Right. The informational self-determination concedes anybody the right to have control about disclosure and use of personal data.

Secrecy of Telecommunications. The secrecy of telecommunications in anchored in Art. 10 of the German Constitution. Referring to Decision [Fed, Decision: 67, 157 (172):1984], Bräunlich *et al.* note that the secrecy of telecommunications protects the confidentiality of individual communication transmitted by means of telecommunications against state intervention.

2.2.2. The Method KORA and the Derivation of Legal Criteria and Technical Design Goals

The development of legally-compliant technology poses a challenge which is to be addressed in an interdisciplinary collaboration. On the one side legal expertise contributes to the understanding of legal provisions while technical expertise supports the enforcement of these provisions by technical constructions. On a scientific level, efforts have been undertaken to conceptualize the development of legally-compliant technology. One result of these efforts is the method KORA (**Ko**nkretisierung **R**echtlicher **A**nforderungen, *engl.*: Concretization of Legal Requirements) [HPR93]. KORA is a four-step method for acquiring technical design proposals based on legal provisions.

1. In the first step, application-specific *legal requirements* are identified from the relevant parts of the constitution, relevant constitutional court decisions, and the opportunities and risks of the technology under investigation.

2. In the second step, legal requirements are made more concrete to so-called *legal criteria* by considering simple law regulations and decisions from other courts.

3. In the third step, a language shift between the legal and technical language happens and technical expertise enters the process. Legal criteria are made more concrete to so-called *technical design goals* in an interdisciplinary dialogue.

4. In the fourth step, a *technical design proposal* is deduced from the design goals. Due to the systematic deduction, this proposal is supposed to be constitutionally compliant.

While the first two steps are driven by legal experts, a language shift happens between KORA step 2 and KORA step 3. The latter steps are consequently driven by technical experts. The method has proven its significance and its value in several applications, *e.g.* the development of mobile devices [HJHL11], the usage of multimedia documents for the approval of new plants according to the Federal Immission Control Act [IL00], and the usage of digital signatures [PR94]. Bräunlich *et al.* [BGRR13] applied the method to derive technical design goals for constitutionally-compliant Internet voting. As a result of their work, Bräunlich *et al.* derived 5 legal requirements, 13 legal criteria, and 30 technical design goals. The complete list of legal requirements, legal criteria, and technical design goals compiled by Bräunlich *et al.* is provided in Appendix A.

It shall be emphasized that the purpose of KORA is to support the interdisciplinary development of legally-compliant technology, rather than the evaluation of technical proposals against legally-founded requirements. Bräunlich *et al.* emphasize this aim by the following statement:

> *"As opposed to the a posteriori, digital legal evaluation in terms of legally-compliant or legally-incompliant, [...], KORA strives for optimizing technical solutions with regard to legal requirements."*

Often, such an ideal development scenario is not given and technology is ahead of legal provisions. Consequently, having a variety of technical solutions at hand, what shall be done is to evaluate technology against technical requirements.

Referring to Bräunlich *et al.*'s work, it turns out that neither legal criteria nor technical design goals might serve as legally-founded evaluation criteria for Internet voting systems. Both legal criteria and technical design goals capture technical requirements, refinements of requirements, and technical measures supporting the enforcement of requirements. Consequently, evaluating Internet voting systems against these criteria or technical design goals might give certain legal provisions unintentionally more weight than others.

Consider for instance the following technical design goals (TDG) determined by Bräunlich *et al.*:

TDG 22: *"Third parties must not be capable of linking a vote to the voter who cast the respective vote."*

TDG 23: *"The voter must not be capable of proving her vote to any third party."*

Technical design goal TDG 22 cannot be enforced without the enforcement of technical design goal TDG 23. If a voter would be capable of proving her vote to a third party, that third party would immediately be capable of linking the vote to the voter who cast it. Consequently, while the technical design goal 22 essentially corresponds to a technical requirement, technical design goal 23 corresponds to a refinement of technical design goal 22.

Having the interdisciplinary research results by Bräunlich *et al.*, one question arises: Can the results of KORA be transformed into technical requirements?

2.3. Determination of Security Requirements

On the basis of legal criteria and technical design goals derived by Bräunlich *et al.* [BGRR13], we derive security requirements for Internet voting systems. Here, particular attention is given to the legal criterion *assurance*, as it conceptually differs from other legal criteria. In his dissertation [Ric12], Richter explicates on the assurance criterion:

> *"Assurance is an instrumental criterion, which ensures the enforcement of all other criteria even in the presence of attacks and failures."*

In the remainder of this chapter, the assurance criterion forms the basis for the separation between security and non-security requirements.

2.3.1. Research Approach

We present our research approach to determine security requirements from legal criteria and technical design goals derived by Bräunlich *et al.*. The approach is clarified by providing an accompanying example, namely the legal criterion *unknowableness*.

Identification of Core Criteria. We disassemble legal criteria into *core criteria* for Internet voting systems, *descriptive refinements* of these core criteria, and *measures* supporting the enforcement of core criteria. By doing so, we make sure that each part of a legal criterion is considered in the identification of core criteria. As a result, we obtain at least one core criterion for each legal criterion. This first step allows one to narrow down comprehensive legal criteria into their essential content; the foundation for technical requirements.

The unknowableness criterion captures one single core criterion:

> *"The content of the cast binding vote must be protected throughout the entire voting phase. [...] It must not be possible to anybody except the voter to read or obtain the voter's cast intention by any other means before the end of the voting phase."*

The core criterion is supported by the following descriptive refinement:

> *"Throughout the tallying phase, the content of cast votes has to be processed. However, prior to vote tallying, the content of cast votes must not be revealed certainly to anybody except the voter who cast that vote. [...] The content of a vote cast in a private, professional, or public context must be protected against being spied out by third parties; being it either by shoulder-surfing, having read access on the voting device, or capturing the communication. There must be effective measures implemented to protect secrecy of the vote in the private sphere. Unknowableness simultaneously protects against undue influence in the moment of vote casting, as well as against the calculation of intermediate results."*

Ultimately, the legal criterion proposes measures for the enforcement of the core criterion:

> *"For that purpose, in polling station voting, cast ballots remain inaccessible throughout the voting phase."*

Relating Core Criteria and Technical Design Goals. We assign the technical design goals derived by Bräunlich *et al.* from legal criteria to the respective core criteria. We therefore determine whether all technical design goals resulting from legal criteria can be related to the identified core criteria. This step serves two purposes: First, the step serves as cross-check for the identification of core criteria, *i.e.* it enables one to see whether all aspects of the technical design goals can be related to core criteria. Second, the step builds the foundation for the next process step, namely the identification of security-related core criteria.

For unknowableness, only one core criterion has been identified. Bräunlich *et al.* derived the following technical design goals from the unknowableness criterion:

> TDG 10: *"The calculation of intermediate results must not be possible."*
>
> TDG 23: *"The voter must not be capable of proving her vote to any third party."*

Technical design goal 10 closely resembles a technical requirement defined for the identified core criterion. Technical design goal 23 represents a refined requirement which supports the enforcement of the identified core criterion. In conclusion, both technical design goals relate to the identified core criterion.

Identification of Security Security-Related Core Criteria. We identify core criteria core criteria that relate to *at least* one technical design goal that Bräunlich *et al.* derive from the instrumental legal criterion assurance. This step supports the identification of security-related core criteria, the basis for technical security requirements.

Both, technical design goals 10 and 23 are derived from the assurance criterion. Consequently, the determined core criterion is a security-related core criterion.

Determination of Technical Requirements. KORA foresees the definition of legal criteria in legal jargon. Given the fact that core criteria are an extract of legal criteria, those core criteria are generally a solid foundation for the specification of technical requirements. The last step of this process consequently transforms core criteria into technical requirements.

The core criterion derived from the legal criterion unknowableness is transformed into the following technical requirement.

Fairness (Security Requirement): The voting system does not provide evidence about any eligible voter's intention before the end of the voting phase.

2.3.2. Execution and Results

We present the execution of our research method and the final result of the execution. A summary of the derivation process and the results is provided in Figure 2.1.

Legal Criterion: Usability

The legal criterion *usability* is refined into three core criteria.

Core Criterion: *"The self-determined vote can only be guaranteed if the voter can use the system according to her intention. [...] Furthermore system usage must either be self-explanatory or has to be introduced to the voter by adequate means throughout the voting phase."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 5: *"The ballot must be neutral."* (No relation to Assurance)
- TDG 14: *"The essential steps of the vote casting process must be understandable to any voter."* (No relation to Assurance)
- TDG 16: *"All voters must obtain the same result with equal usage."* (No relation to Assurance)
- TDG 18: *"The vote may only be cast and stored after a confirmation by the voter."* (No relation to Assurance)
- TDG 20: *"All voters must receive a message regarding the (non-)success of her voting process."* (No relation to Assurance)

No technical design goal relates to the assurance criterion. Hence, we define the following non-security requirement.

System Usability (Non-Security Requirement): The voting system is usable to all eligible voters.

Core Criterion: *"For the sake of implementing the voter's self-realization, each voter must have the possibility to vote according to her intention, to abstain from the election, or to cast an invalid vote."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 11: *"The calculation of the election result must be started after the official voting phase by members of the election committee."* (Relation to Assurance)

- TDG 19: *"It must be ensured that the vote is correctly transmitted."* (Relation to Assurance)
- TDG 21: *"A voting note must only be taken after a binding vote has been cast."* (Relation to Assurance)
- TDG 24: *"It must not be possible to manipulate the stored binding votes."* (Relation to Assurance)
- TDG 25: *"The system must compute the correct result."* (Relation to Assurance)
- TDG 26: *"It must not be possible to manipulate the election result."* (Relation to Assurance)

All technical design goals relate to the assurance criterion. We consequently establish the following security requirement.

Vote Integrity (Security Requirement): The voting system ensures that each vote is correctly included in the election result.

Core Criterion: *"The personal vote casting must be largely guaranteed also to handicapped voters."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 15: *"All voters must be able to conduct the vote casting process."* (No relation to Assurance)

The technical design goal does not relate to the assurance criterion. We consequently establish the following non-security requirement.

System Accessibility (Non-Security Requirement): The voting system is accessible to all eligible voters.

Legal Criterion: Availability

The legal criterion *availability* is refined into three core criteria.

Core Criterion: *"All relevant election data, such as authentication data, the electoral register, and the list of candidates, must be available and up to date throughout the entire election, such that eligible voters can participate in the election self-determined and equally."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 1: *"Unauthorized parties must not have the possibility to view voter data."* (Relation to Assurance)
- TDG 2: *"Unauthorized parties must not have the possibility to manipulate voter data."* (Relation to Assurance)
- TDG 7: *"The election committee must start the election at the predetermined time."* (Relation to Assurance)
- TDG 9: *"The election committee must stop the election at the predetermined time."* (Relation to Assurance)
- TDG 11: *"The calculation of the election result must be started after the official voting phase by members of the election committee."* (Relation to Assurance)
- TDG 12: *"Only eligible voters may access successfully the Internet voting system."* (Relation to Assurance)
- TDG 21: *"A voting note must only be taken after a binding vote has been cast."* (Relation to Assurance)

All technical design goals relate to the assurance criterion. We consequently establish the following security requirement.

Voter Availability (Security Requirement): The voting system does not exclude eligible voters from casting their intention.

Core Criterion: "The voting system itself has to be available throughout the entire election phase without major failures, such that votes can be processed."

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 8: *"After a system failure, it must be possible to resume the election."* (Relation to Assurance)

The technical design goal does relate to the assurance criterion. We consequently establish the following security requirement.

System Availability (Security Requirement): The voting system is available to all eligible voters at any point in time.

Core Criterion: "For an obligatory Internet election, all eligible voters must have physical access to the voting system."

Technical Design Goals: The following technical design goal can be related to the determined core criterion.

- TDG 17: *"Eligible voters must have the possibility to cast votes at any time of the voting phase."* (No relation to Assurance)

The technical design goal does not relate to the assurance criterion. We consequently establish the following non-security requirement.

System Reachibility (Non-Security Requirement): The Internet voting system is physically accessible to all eligible voters.

Legal Criterion: Equality of Votes

The legal criterion *equality of votes* is refined into one core criterion.

Core Criterion: *"The voting system has consequently to be set up such that it only accepts votes of eligible voters and only accepts these votes once and with equal weight."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 13: *"Eligible voters may cast only and exactly one binding vote."* (Relation to Assurance)
- TDG 19: *"It must be ensured that the vote is correctly transmitted."* (Relation to Assurance)
- TDG 25: *"The system must compute the correct result."* (Relation to Assurance)

All technical design goals relate to the assurance criterion. We consequently establish the following security requirement.

Eligibility (Security Requirement): The voting system ensures that only eligible voters' votes are included once in the election result.

Legal Criterion: Neutrality

The legal criterion *neutrality* is refined into one core criterion.

Core Criterion: *"A content-related influence of voters because of the Internet voting system must be prevented."*

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 5: *"The ballot must be neutral."* (No relation to Assurance)
- TDG 6: *"Unauthorized parties must not have the possibility to change the ballot data."* (Relation to Assurance)

At least one of both technical design goals relates to the assurance criterion, such that the following security requirement is established.

System Neutrality (Security Requirement): The voting system does not influence the eligible voter's intention.

Legal Criterion: Unknowableness

The legal criterion *unknowableness* is refined into one core criterion.

Core Criterion: "The content of the cast binding vote must be protected throughout the entire voting phase.[...] It must not be possible to anybody except the voter to read or obtain the voter's cast intention by any other means before the end of the voting phase."

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 10: *"The calculation of intermediate results must not be possible."* (Relation to Assurance)
- TDG 23: *"The voter must not be capable of proving her vote to any third party."* (Relation to Assurance)

All technical design goals relate to the assurance criterion. We consequently establish the following security requirement.

Fairness (Security Requirement): The voting system does not provide evidence about any eligible voter's intention before the end of the voting phase.

Legal Criterion: Unlinkability

The legal criterion *unlinkability* is refined into one core criterion.

Core Criterion: "At no point in time, it must be possible to link the content of cast binding votes to the real identity of the voter who cast that vote."

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 22: *"Third parties must not be capable of linking a vote to the voter who cast the respective vote."*
- TDG 23: *"The voter must not be capable of proving her vote to any third party."* (Relation to Assurance)

All technical design goals relate to the assurance criterion. We consequently establish the following security requirement.

Vote Secrecy (Security Requirement): The voting system does not provide more evidence about a specific eligible voter's intention than the election result does.

Legal Criterion: Individual Control

The legal criterion *individual control* is refined into one core criterion.

Core Criterion: "Each voter must be able to control that the system stores and tallies the vote with the voter's intention."

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 4: *"Any voter must have the possibility to view and influence both extent and purpose of stored her personal data."* (No relation to Assurance)
- TDG 27: *"Any voter must be able to verify that her vote has been included in the election result."* (No relation to Assurance)
- TDG 29: *"The election must be logged."* (Relation to Assurance)
- TDG 30: *"The election data must be archived in a traceable and evidence-proven manner."* (Relation to Assurance)

Technical design goals 29 and 30 are not directly in place to deploy the core criterion 11. Rather, these goals ensure that generated election logs and election archives are protected from malicious access. As such core criterion 11 is *not* considered a security requirement.

Individual Verifiability (Non-Security Requirement): The voting system offers each eligible voter the possibility to verify that her intention has been correctly included in the election result.

Legal Criterion: Public Control

The legal criterion *public control* is refined into one core criterion.

Core Criterion: "Any citizen must be able to control the constitutionally-compliant process of any vote casting, in other words to control the enforcement of the principles universal, direct, equal, free, and secret elections."

Technical Design Goals: The following technical design goals can be related to the determined core criterion.

- TDG 28: *"The public must be able to verify that the election result has been derived correctly."* (No relation to Assurance)
- TDG 29: *"The election must be logged."* (Relation to Assurance)

- TDG 30: *"The election data must be archived in a traceable and evidence-proven manner."* (Relation to Assurance)

Technical design goals 29 and 30 are not directly in place to deploy the core criterion 12 and as such core criterion 12 is *not* considered a security requirement.

Public Controllability (Non-Security Requirement): The voting system offers any observer the possibility to control that all technical requirements resulting from the principles universal, direct, equal, free, and secret elections are enforced.

Legal Criterion: Data Economy

The legal criterion *data economy* is refined into one core criterion.

Core Criterion: *"The voting system shall only request and store the personal data without which the system does not operate correctly."*

Technical Design Goals: The following technical design goal can be related to the determined core criterion.

- TDG 3: *"Only data required shall be stored."* (No relation to Assurance)

The technical design goal does not relate to the assurance criterion. We consequently establish the following non-security requirement.

Data Minimization (Non-Security Requirement): The voting system shall only request and store the personal data without which the system does not operate correctly.

Legal Criterion: Data Transparency

The legal criterion *data transparency* is refined into one core criterion.

Core Criterion: *"The voting system shall offer the voter a possibility to view the personal data about herself stored and processed by the system."*

Technical Design Goals: The following technical design goal can be related to the determined core criterion.

- TDG 4: *"Any voter must have the possibility to view and influence both extent and purpose of stored her personal data."* (No relation to Assurance)

Technical design goal 4 does not relate to the assurance criterion. We consequently define the following non-security requirement.

Data Inspection (Non-Security Requirement): The voting system shall offer the voter a possibility to view the personal data about herself stored and processed by the system.

Legal Criterion: Appropriation

The legal criterion *appropriation* is refined into one core criterion.

Core Criterion: *"The voting system shall only process personal data without which the system does not operate correctly."*

Technical Design Goals: The following technical design goal can be related to the determined core criterion.

- TDG 1: *"Unauthorized parties must not have the possibility to view voter data."* (Relation to Assurance)
- TDG 3: *"Only data required shall be stored."* (No relation to Assurance)

Technical design goal 1 relates to the assurance criterion. We consequently define the following security requirement.

Data Access Protection (Security Requirement): The voting system shall prevent unauthorized parties from viewing voter data.

In the remainder of this work, voter data are data that can be directly related to a voter identity.

Legal Criterion: Data Controllability

The legal criterion *data controllability* is refined into one core criterion.

Core Criterion: *"The Internet voting system shall shall offer the voter a possibility to execute [cancellation, rectification, and blocking] rights."*

Technical Design Goals: The following technical design goal can be related to the determined core criterion.

- TDG 4: *"Any voter must have the possibility to view and influence both extent and purpose of stored her personal data."* (No relation to Assurance)

Technical design goal 4 does not relate to the assurance criterion. We consequently define the following non-security requirement.

Data Adaptation (Non-Security Requirement): The voting system shall offer the voter the possibility to adapt her personal data.

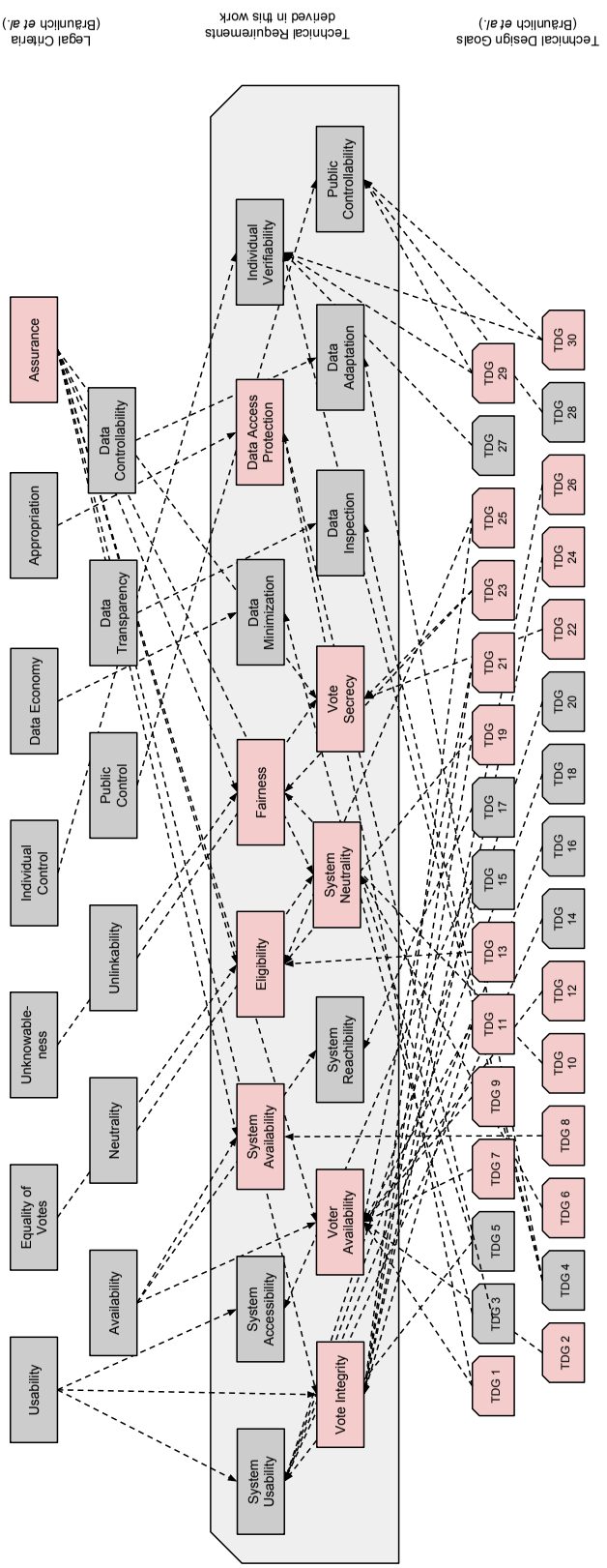

Figure 2.1: On the basis of Bräunlich *et al.*'s [BGRR13] legal criteria and technical design goals, 16 technical requirements have been derived. Stemming from the legal criterion *assurance*, security requirements are highlighted in red. The complete list of legal requirements, legal criteria, and technical design goals compiled by Bräunlich *et al.* is provided in Appendix A.

2.4. Summary

On the basis of preliminary research conducted by Bräunlich *et al.* [BGRR13], technical requirements for Internet voting systems were determined. In the remainder of this work, we restrict our focus to security aspects. We therefore present the resulting list of security requirements for Internet voting systems in alphabetical order:

- Data Access Protection: The voting system shall prevent unauthorized parties from viewing voter data.
- Eligibility: The voting system ensures that only eligible voters' votes are included once in the election result.
- Fairness: The voting system does not provide evidence about any eligible voter's intention before the end of the voting phase.
- System Availability: At any point in time, the voting system is available to all eligible voters.
- System Neutrality: The voting system does not influence the eligible voter's intention.
- Vote Integrity: The voting system ensures that each vote is correctly included in the election result.
- Vote Secrecy: The voting system does not provide more evidence about a specific eligible voter's intention than the election result does.
- Voter Availability: The voting system does not exclude eligible voters from casting their intention.

When constructing Internet voting systems, developers should take these requirements into account and tailor systems towards them. However, Internet voting systems –as any other voting method– cannot enforce simultaneously all security requirements to their full extent. Improving one voting system with regard to specific security requirements often comes at the cost of reducing the enforcement of other security requirements. From the legal point of view, the legal latitude allows the legislator to constrain the satisfaction of certain constitutional principles in favor of others [Fed, Decision: 59, 119 (124):1981].

Bearing the determined security requirements for Internet voting systems and the legal latitude in mind, the goal of the following chapter is the construction of a security evaluation framework for Internet voting schemes, the conceptual underpinning of Internet voting systems. On the one side, an evaluation framework shall incorporate the impact caused by conducting specific attacks with regard to specific security requirements[3]. On

[3]The maximum impact depends on the security requirement under investigation: for instance, vote secrecy can only be violated for voters who cast their binding vote, while an adversary might illegitimately access voter data of all eligible voters (violation of data access protection).

the other side, given that Internet voting schemes enforce security requirements by building upon assumptions about their environment, it is reasonable to evaluate Internet voting schemes not only with regard to the assumptions but furthermore also with regard to the criticality of the assumptions within the target election setting. This idea will be the guideline for the construction of a security framework presented in the following chapter.

Chapter 3

Construction of a Security Evaluation Framework for Internet Voting Schemes

The legally-founded security requirements for Internet voting systems have been determined in the previous chapter. On the basis of these requirements, the goal of this chapter is the construction of a security evaluation framework for Internet voting schemes, the conceptual underpinning of Internet voting systems.

In the first part of this chapter, we specify Internet voting schemes as core of Internet voting systems and adapt the derived security requirements for Internet voting systems accordingly. In the second part, we review related work and contrast them to our contribution. We subsequently provide the foundations of the security evaluation framework. The fourth part of this chapter is dedicated to the actual construction in terms of building blocks and and processes. Thereafter, we provide short guidelines for the deduction of qualitative security models and for the determination of election settings. We subsequently evaluate the framework with regard to four properties borrowed from the field of measure theory. A summary of this chapter is given in the last section.

Parts of this chapter have been published in the journal *Datenschutz und Datensicherheit* [2] and in the journal *Annals of Telecommunications* [22].

3.1. Internet Voting Schemes and their Security Requirements

We first specify the target of our evaluation framework, namely Internet voting schemes, the core of Internet voting systems. Subsequently, the security requirements established in Section 2 are revised and tailored towards Internet voting schemes.

3.1.1. Internet Voting Schemes

From a legal perspective, elections in their entirety have to be conducted in a legally-compliant way. The conduct of Internet-based elections incorporates several dimensions such as cryptographic protocols, the hard-/ and software implementing and running the protocols, and authorities in charge of administrating hard-/ and software components.

Throughout this thesis, we restrict our focus to Internet voting schemes. To obtain a precise understanding about the specification of an Internet voting scheme, we delineate Internet voting schemes from the remaining parts of the Internet voting system.

Several works propose reference frameworks for electronic voting systems [Lun10, Sch04]. Because of the fact that Schryen's framework directly refers to Internet voting systems, we select that framework as foundation for our specification. An overview of the framework is provided in Figure 3.1. The core of an Internet voting system is the *organization dimension*.

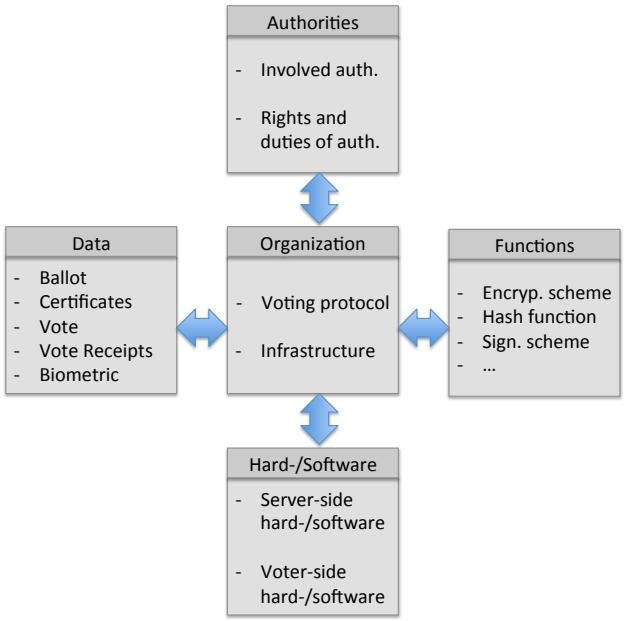

Figure 3.1: Reference framework for electronic voting systems according to Schryen [Sch04]. The core of an Internet voting system is the organization dimension including the protocol and the infrastructure of the system.

This dimension captures the fundamental protocol of the Internet voting system, which prescribes how data is processed and exchanged by different components involved in the voting process. Additionally, the organization dimension covers infrastructural aspects in terms of how many servers are available to conduct the election.

According to the reference framework, there are four dimensions surrounding the organization dimension. These are the *data* dimension, the *hardware and software* dimension, the *functions* dimension, and the *authorities* dimension: The data dimension captures contents such as the ballots to be used throughout the election, certificates to establish trust between the components involved in the election, the votes, vote receipts, and optionally biometric authentication data. The functions dimension captures the way cryptographic components are implemented, *i.e.* which cryptographic algorithms are in place. The dimension might for instance prescribe what what type of encryption scheme and what type of cryptographic hash functions are used. The hardware and software dimension captures

which type of hardware and software is used at both the voter side and the server side. This dimension might for instance prescribe that only certified hardware and/or software is used. The authorities dimension determines which authorities are involved in the Internet based elections, *i.e.* which authorities take which protocol role, and provide and manage which kind of infrastructure.

In line with the reference framework, we consider the organization dimension as the core of Internet voting systems. We consequently define an Internet voting scheme as the organization dimension of an Internet voting system. This means that Internet voting schemes capture the components involved in the voting process and their respective roles in generating, exchanging and processing election data. Such components might be central voting providers and voters' voting devices.

By its specification, the organization dimension closely relates to the data dimension. In fact, it turns out that the feasibility and security of different Internet voting schemes might depend on the data dimension. For instance, specific voting protocols can only be applied when the ballot complexity is low [CPP13, Joa14]. Throughout the scheme evaluation, we therefore consider the organization dimension together with the data dimension, *i.e.* Internet voting schemes are evaluated as applied for one specific election.

With the advance and practical application of Internet voting, the interdependence of the organization dimension and the voter interacting with the voting protocol and infrastructure has become more and more apparent. In accordance to legal provisions, *e.g.* Bräunlich *et al.* [BGRR13], and Madise and Vinkel [MV11], recent research tends to consider the voter as part of the voting protocol, see for instance the work by Carlos *et al.* [CMPC13]. Following this tendency, in addition to voting providers and voting devices, we consider the voters as part of an Internet voting scheme.

A reference Internet voting scheme is provided in Figure 3.2. An Internet voting scheme is composed of several *components* and *channels* between these components. According to Schryen's reference framework, components build the infrastructure, which take specific roles of the protocol, *e.g.* the voter, the voting device, or central voting servers.

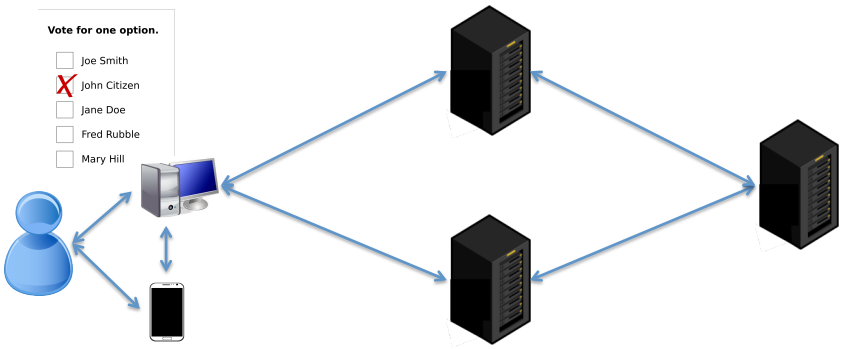

Figure 3.2: Reference Internet voting scheme: Each voting scheme prescribes different roles between which the election is conducted. In addition to the interaction between central voting servers and voter-side systems, the voter directly interacts with voter-side systems.

3.1.2. Security Requirements of Internet Voting Schemes

The security requirement *system availability* requires that the hardware and software components involved in the election process are able to provide their service throughout the entire election. Ensuring the availability of these components does, however, depend on the hard-/software dimension, *i.e.* the hardware and software in place. We therefore do not consider system availability throughout the evaluation of Internet voting schemes.

We make the general assumption that anybody –including voters and the public– verify everything that they can verify and raise a complaint in case verification fails[4]. We additionally make the assumption that ballots are published in advance to the election and voters shape their vote intention prior to the actual voting process. Consequently, if voters get presented an altered ballot throughout the voting phase, they detect this discrepancy and raise a complaint to the election officials. We therefore do not consider *system neutrality* throughout the evaluation of Internet voting schemes.

Analogously to the case of system neutrality, if the system does illegitimately exclude voters from casting their intention, the voter raises a complaint, thereby triggering further investigation. We therefore do not consider *voter availability* throughout the evaluation of Internet voting schemes.

3.2. Related Work

We review related works on the security evaluation of Internet voting schemes and Internet voting systems. We subdivide these works into qualitative approaches, namely the Common Criteria for IT-security evaluation and resilience term evaluation, and quantitative approaches, namely threat tree evaluation and quantitative evaluation, and quantification approaches for qualitative evaluations.

Common Criteria for IT-Security Evaluation

The Common Criteria for Information Technology Security Evaluation are an international standard for information security. The development of the Common Criteria is advanced by the states Australia / New Zealand, Canada, France, Germany, Japan, the Netherlands, Spain, the United Kingdom, and the United States of America. Version 1.0 of the Common Criteria has been released in 1996. Subsequently, the Common Criteria have been captured within the ISO standard 15408 in 1999. The most recent version of the Common Criteria is version 3.1 release 4. The Common Criteria incorporate the concept of Protection Profiles. Protection Profiles capture security requirements for generic end products. Protection Profiles are deliberately abstract and independent of concrete products. As such, Protection Profiles generally address product groups. Consequently,

[4]For research on the voter motivation and usable verifiability, we refer the reader to the works by Olembo *et al.* [ORBV14] and Budurushi *et al.* [BWV14]

Protection Profiles are created in representation of the end users. Protection Profiles allow developers to fall back on established security requirements. As such, the evaluation and certification according to Protection Profiles is a valuable and confidence-building measure for developers. Numerous Protection Profiles have been developed for a diversity of security-critical products; among them there are Protection Profiles for secure signature-creation devices [KLP+01], sovereign documents [Bun09], and health-care products [KGK09]. Several Protection Profiles have been developed specifically for electronic voting technologies, *e.g.* PP-CIVIS [Sec06], IEEE P1583 [IEE05], Karokola *et al.* [KKY12] and Lee *et al.* [LLWK10]. Furthermore, one Protection Profile for Internet voting systems has been developed [VV08].

In spite of their longstanding history, the Common Criteria turn out to be inappropriate as evaluation framework for this work. Given the partial opposing nature of legally-founded security requirements, those requirements can only be enforced under certain assumptions about the environment. The Common Criteria opens the possibility to specify assumptions. However, first, the enforcement and evaluation of security assumptions within the operational environment is not part of the Common Criteria. Hence, those assumptions might be unreasonably high. In that case, the evaluation (and potential certification) of Internet voting systems according to the Protection Profile might be questionable. Second, according to the legal latitude (see Section 1.1), there is not *one* specific set of assumptions about the operational environment. Rather, systems might be considered *compliant* if legal provisions or refined security requirements are enforced in a balanced way tailored towards the election setting. Consequently, single Protection Profiles do not sufficiently incorporate the concept of legal latitude; hence, legal provisions cannot be adequately represented within Protection Profiles.

Resilience Term Evaluation

Volkamer and Grimm [VG09] were motivated by the fact that established evaluation frameworks for Internet voting systems remain abstract and do not adequately consider trust distribution concepts, such as separation of duties and multiplicity of functions. To address this shortcoming, the authors developed the concept of resilience terms. These terms allow one to capture complex trust distributions and to express which entities have to be trusted - in particular not to collaborate maliciously - in order to fulfill security requirements. Resilience terms are specified as follows:

- A system is called k-resilient with regard to a security requirement if at least k entities out of the set of all entities must be trusted not to collaborate maliciously in order to violate the respective security requirement.
- A system is called k-out-of-N-resilient if at least k entities out of the set of entities N must be trusted not to collaborate maliciously in order to violate the respective security requirement.

- A system is called $(k_1 + \cdots + k_m)$-out-of-$(N_1, \ldots, N_m)$-resilient if at least k_1 entities out of the set of entities N_1 and ... and at least k_m entities out of the set of entities N_m must be trusted not to collaborate maliciously in order to violate the respective security requirement.

- A system is called $(k_{11} + \cdots + k_{m1})$-out-of-$(N_{11}, \ldots, N_{m1})$, ..., $(k_{1n} + \cdots + k_{mn})$-out-of-$(N_{1n}, \ldots, N_{mn})$-resilient if at least k_{11} entities out of the set of entities N_{11} and ... and at least k_{m1} entities out of the set of entities N_{m1} or ... or at least k_{1n} entities out of the set of entities N_{1n} and ... and at least k_{mn} entities out of the set of entities N_{mn} must be trusted not to collaborate maliciously in order to violate the respective security requirement.

Because of their formal structure, resilience terms build a precise specification language for assumptions upon which Internet voting systems are based. Hence, resilience terms build the foundation for the evaluation of assumptions as part of the evaluation of Internet voting schemes.

While the concept of resilience terms overcomes one shortcoming of the Common Criteria, it falls short for the following concerns: Resilience terms evaluate and express the security of Internet voting systems with regard to trust distributions, *i.e.* which entities need to be trusted not to collaborate maliciously in order to enforce security requirements. These trust distributions do, however, not incorporate the election setting into the security evaluation and expression. Given the potential complexity of resilience terms, the identification of an adequate Internet voting system for their election setting easily overwhelms the election official. Furthermore, resilience terms remain abstract in the following sense: The concept of resilience terms is tailored towards a possibilistic interpretation of security, as it only captures collaborations by central entities of the system. The security evaluation and expression of Internet voting system with regard to central entities might be too restrictive, because adversaries might consider other attack targets to violate security requirements, for instance targets on the voter side, such as voting devices or influencing voters throughout the vote casting process.

Threat Trees and Quantitative Evaluation

Several works have addressed the assessment of risks for electronic voting systems [PYL10, BC07, PLY11, Lau04, NK06, KN08, BM07] by deriving threats trees for these systems. Comprehensive threat trees for electronic voting (or Internet voting) systems are of great value for the deduction of adversaries violating security requirements. Yet, the fine-grained threats considered in these works require decision makers to assign probabilities to specific threats. Reviewing threat trees for Internet voting systems poses a significant burden on election officials, *e.g.* [EAC09] provides a 18-page threat tree for Internet voting. Additionally, given the unstructured nature of threats, estimating the severity of threats generally

exceeds the expertise of election officials. Pardue *et al.* [PLY10] support the interpretation of complex threat trees by incorporating Monte-Carlo simulations. To evaluate an Internet voting system, a system analyst estimates (with uncertainty) the probability with which an adversary exercises specific attacks and the impact caused by those attacks. In a second step, the previous estimates are adjusted by (uncertain) estimates regarding the attacker's motivation to exercise specific attacks and the complexity of specific attacks. While this approach facilitates the interpretation of large and complex threat trees, the approach is tailored towards system analysts. Hence, the approach does not foresee the incorporation of election settings by election officials. Ouchani *et al.* [OJM11] quantify attack patterns of the Common Attack Pattern Enumeration and Classification catalogue[5]. Luna *et al.* [LSK12] develop a quantitative threat modeling with particular focus on privacy-by-design requirement. Their approach derives from the Microsoft threat modeling approach STRIDE and threat-risk ranking approach DREAD as well as established privacy protection goals. Given the fine granularity of attack patterns (Ouchani *et al.*) and threats (Luna *et al.*), their quantification might easily overwhelm election officials when identifying the most appropriate Internet voting system for their election setting. Vejačka [Vej13] quantitatively evaluates the Estonian, the Washington D.C., and the Edmonton Internet voting systems with regard to 14 requirements. Fundamental idea of the quantification approach are the importance weighting of the established requirements and the qualitative evaluation of the schemes. Both the weighting of requirements and the qualitative evaluation of Internet voting schemes remain abstract such that presented approach cannot be transferred to other contexts. Li *et al.* [LKZ14] develop a taxonomy for Internet voting schemes. To achieve their goal, the authors evaluate 14 Internet voting schemes with regard to 12 technical requirements. The evaluation remains abstract, as it only provides short arguments about whether the different schemes satisfy or do not satisfy the target requirements. Jonker *et al.* [JMP09] propose a framework for the quantification of voter privacy in the presence of conspiring voters. On the foundation of formal methods, their approach measures to what extent voters are capable of cooperating with the voter in order to leak knowledge (not necessarily the entire knowledge) about their voting decision. On the foundation of legally-founded security requirements, we consider any knowledge beyond publicly available knowledge about a specific voter's vote a secrecy violation.

Küsters *et al.* [KTV11, KTV12] provide a formal framework for measuring the level of verifiability, privacy, coercion-resistance, and accountability of voting protocols. The framework measures (by means of a so called δ) the adversary's chance of achieving her goal, *e.g.* making a verifier accept an incorrect election result (*verifiability*) or distinguishing between the fact whether an observed voter casts a vote for one candidate or another candidate (*privacy*). The measurement depends on a number of factors, such as the set of honest authorities, the number of honest voters, the number of voting options, and

[5]Refer to `https://capec.mitre.org`

probability distributions for these voting options. In other words, the framework precisely measures to what extent specific adversarial capabilities (given in terms of dishonest authorities and voters) suffice to cause specific impact on a specific requirement. In spite of its contribution, the framework does not provide an interface to election officials and does not incorporate election settings (*e.g.* by means of probabilistic adversaries). Hence, the framework does not directly support election officials in evaluating a scheme's adequacy within concrete election settings. To that end, both works turn out to have complementary goals. Despite this difference, both works address quantitative security from different directions and can therefore benefit from each other. We consequently foresee an integration of both approaches as future work.

Quantification of Qualitative Security Evaluations

On the foundation of resilience terms [VG09], Schryen *et al.* [SVRH11] develop a quantitative trust metric upon propositional logic. As foundation for their quantification, the authors determine resilience terms for security requirements in distributed systems. Thereafter, they compute the probability that security requirements might be violated on the basis of failure probabilities of individual entities. The quantification builds on standard probability theory. While the quantification of resilience terms is a reasonable approach, the approach inherits one essential shortcoming of the resilience term evaluation, namely the fact that the evaluation focuses on central entities of the voting system. Furthermore, the quantification process falls short because of the fact that the authors remain unclear about how election settings are to be incorporated into the quantification process, *e.g.* how uncertainty is exactly handled. Similar to Schryen *et al*, Lazarus *et al.* [LDEH11] construct a quantitative threat evaluation by means of the metric *attack team size*. The metric measures how many entities are knowingly involved in attacks targeting at different types of security requirements. In spite of its clarity, the attack team size metric might oversimplify in that regard that it considers the vulnerability of entities to be equally weighted or equally weighted between insider and outsider attackers. Consequently, the approach might not consider the election setting adequately and consequently not evaluate security adequately within specific settings.

The gained insights reveal the lack for a security evaluation framework that on the side side precisely captures conceptual shortcomings of Internet voting schemes with regard to legally-founded security requirements, and on the other side evaluates these shortcomings within specific election settings.

3.3. Foundations of the Security Evaluation Framework

Before diving into the details of its construction, we provide the necessary foundations of the security evaluation framework. In the first part of this section, we determine properties

that the security evaluation framework shall possess. We subsequently introduce the reader into the basics of measurement theory. Afterwards, we present the technique of Monte-Carlo simulations, an approach to numerically evaluate complex stochastic models. Thereafter, we introduce the concept of Pareto dominance.

3.3.1. Properties of the Security Evaluation Framework

Given the fact that no voting method enforces the deployment of legal provisions to their full extent, an Internet voting scheme's benefits and drawbacks within specific election settings must be measurable. Measuring the enforcement of legally-founded security requirements within specific election settings lays the foundation for comparing the legally-founded security of different Internet voting schemes. Before providing the actual construction, we have to determine properties that the intended security evaluation framework shall possess. By its very nature, the framework closely relates to the mathematical concept of a *measure* (refer for instance to Salamon [Sal16]). We therefore base the properties for the construction upon the properties of a measure and adapt them to our context.

The first property a measure must possess is that it must assign the *empty set* of the σ-algebra in the measure space, the measure 0. Transferring this property to the context of security evaluation, we derive two properties: First, the construction must return the quantitive security evaluation result 1, if the adversary has no capabilities. We refer to this property as *no capabilities – perfect security*.

No Capabilities – Perfect Security. If the Internet voting scheme under investigation faces an adversary that has no capabilities, then the quantitative security evaluation result must be 1, unless the security requirement can be violated without any adversarial capabilities[6].

The second property we derive requires that an adversary with specific capabilities cannot cause harm to the Internet voting scheme, if the scheme is resistant against those capabilities. We refer to this property as *capability resistance*.

Capability Resistance. If the Internet voting scheme under investigation proves to be resistant against specific adversarial capabilities, then for any two adversaries that differ only with regard to that capability, the quantitative security evaluation results must be equal against both adversaries.

The second property a measure must possess is *continuity*. In measure theory, the property of continuity is defined by stating that 1) the measure of the infinite union of

[6]This holds for instance true if vote secrecy is not required and the Internet voting scheme under investigation publishes the relation between a voter and her vote.

a sequence of increasing sets $(E_n)_{n\in\mathbb{N}}$ converging towards a set E from the σ-algebra is equal to the measure of E, and 2) the measure of the infinite intersection of a sequence of decreasing sets $(E_n)_{n\in\mathbb{N}}$ converging towards a set E from the σ-algebra is equal to the measure of E. Transferring this property to the context of security evaluation, the property requires that two adversaries can always be found of which one is stronger than the other, such that their quantitative security evaluation results get arbitrarily close to each other. Analogously to measure theory, we refer to this property as *continuity*.

Continuity. If the Internet voting scheme under investigation faces two adversaries that differ arbitrarily little in their capabilities, then also the quantitative security evaluation results must differ arbitrarily little.

The third property a measure must possess is *monotonicity*. In terms of measure theory, the property requires that the measure of a subset of another set from the σ-algebra should be smaller than the measure of the set. The fourth property a measure shall must possess is *σ-additivity*. In terms of measure theory, the property requires that the measure of a union of disjoint subsets of the *σ-algebra* equals the sum of the measure of the disjoint subsets. Both properties are transferred to the context of security evaluation for Internet voting schemes. The resulting property requires that for any two adversaries of which one is stronger than the other, the quantitative security evaluation result of the stronger adversary must be smaller than the quantitative security evaluation result of the other. We refer to the property as *monotonicity*.

Monotonicity. If the Internet voting scheme under investigation faces two adversaries, of which one is stronger than the other, then the quantitive security evaluation results of the scheme must be larger when facing the weaker adversary.

3.3.2. Scales of Measurement

In his seminal work, Stevens [Ste46] determines four types of measurement scales, namely nominal, ordinal, interval, and ratio scales. The differences between these scales types are relevant to the herein constructed security evaluation framework. We therefore describe the measurement scales and highlight differences between them.

Nominal scale. Nominal scales provide categories that do not relate to each other. Such categories might be numbers, attributes, or any other kind of (not necessarily unique) identifier. Categories do not relate to each other such that neither an order of categories, nor any kind of differences or ratio can be defined. The only valid operation on variables that map on nominal scales is to check whether two variables are equal. For instance, eye colors are a nominal scale. For the sake of simplicity, we assume that humans have only

the eye colors blue and brown. While it can be determined whether two random humans have the same or a different eye color, there is no eye color that is larger or smaller than the other one, nor any additive or multiplicative relation between both eye colors.

Ordinal scale. Ordinal scales provide categories that can be related by their order. In the case of ordinal scales, in addition to the equality check for categories, categories can be ranked. However, there is no meaning for the additive or multiplicative relation between two categories. While two students might either have the same grade or one student has a better grade than the other one, there is no additive or multiplicative meaning between two different grades.

Interval scale. As opposed to nominal and ordinal scales, variables that map on interval scales are continuous variables. Consequently, in addition to equality tests and a full ordering of variables, also additive differences between variables have a meaning, *i.e.* the interval size on the interval scale has a meaning. Consider for instance temperature as an instantiation of an interval scale. The difference between 10 °C and 20 °C is equal to the difference between 20 °C and 30 °C. On the other side, there is no multiplicative relation between items mapping on the interval scale, *e.g.* one cannot say 20 °C is twice as warm as 10 °C as the example of the temperatures −1 °C and 2 °C clarifies.

Ratio scale. Ratio scales extend the expressiveness of interval scales by a meaning of the multiplicative relation between variables mapping on these scales. Ratio scales have a precisely defined value *zero*, on the basis of which multiplicative relation can be expressed. For instance, height represents a ratio scale, with zero value 0. Starting from 0, it makes sense to say two inches are twice as high as one inch.

If the security of Internet voting schemes could be compared with regard to one single requirement, ordinal scales would be sufficient. However, in many cases, comparing two schemes with regard to several requirements requires to compare and balance differences in the schemes' enforcement of the requirements. To tailor the security evaluation framework for the comparison of schemes' enforcement of different security requirements, we require the security evaluation framework to map Internet voting schemes on interval scales.

3.3.3. Monte-Carlo Simulations

We introduce the concept of Monte-Carlo simulations to handle uncertainties in the specification of election settings. Consider a mathematical M model that processes several input variables to produce a certain output. If the input variables are not fixed in advance but are rather uncertain, then the mathematical model has to be evaluated in a preferably comprehensive manner with regard to the uncertain variables. If the number of input variables is small, the evaluation might be conducted combinatorially in a deterministic

manner. With an increasing number of uncertain input variables, the evaluation faces the *"curse of dimensionality"*. Hence, the evaluation can no longer be addressed by deterministic means, but rather stochastic approaches are needed. Among the most established approaches, there are Monte-Carlo simulations [MU49]. We provide the fundamentals of Monte-Carlo simulations in the following paragraphs. The description of Monte-Carlo simulations is based upon Raychaudhuri's work [Ray08], and the work by Driels and Shin [DS04].

Determining Input Distributions. In the most general case, data points according to a certain probability distribution are given, while their statistical distribution is unknown. A survey on approaches which allow one to determine the probability distribution from a set of data points is provided by Myers [Mye90]. Among the most prevalent approaches, there is the maximum likelihood estimation (MLE) (refer for instance to [Myu03]) which is briefly summarized. Let independent data points $x_1, \ldots, x_n$ be drawn according to an unknown probability distribution. Let f be the joint probability density function of $x_1, \ldots, x_n$ under a probability density function given in terms of parameters p. Then, the likelihood function lik captures the probability that the data points are drawn according to the density function parameters p. Formally, this can be written as:

$$lik(p) = f(x_1, \ldots, x_n|p)$$

From the domain of possible parameters, the parameter p_{max} is determined that maximizes the logarithmic likelihood function, i.e.

$$p_{max} = \max_p(ln(lik(p))) = \max_p \sum_{i=1}^{n} ln(f(x_i|p))$$

Eventually, finding the most fitting density function parameters can be done by optimization techniques [MBT14].

If no data points are given in advance to the model simulation, the simulation has to be built upon reasonable distribution estimations. Such estimations might be based upon expert knowledge of similar problems. Typical distributions for input variables might be Gaussian, Normal, or Student's t-distribution.

Generating Random Data. Let the random variable X to be sampled be defined by the probability density function f. Let F be the invertible cumulative probability density function of f, and F^{-1} be the inverse of F. We assume a sampler for the uniform distribution U$[0, 1]$. Then, the random variable X can be sampled as follows:

1. Determine z by sampling U$[0, 1]$
2. Compute x by evaluating $F^{-1}(z)$

Simulating the Model. Once the probability distributions of input variables have been determined, the main part of Monte-Carlo simulations is initiated. Let $\mathbb{P}_1, \ldots, \mathbb{P}_k$ be the probability distributions of k insecure input variables. Let n vectors be given:

$$v_1, \ldots, v_n$$

Each vector v_j captures one sample for all input variables

$$v_j = (i_1, \ldots, i_k) \text{ with } i_1 \leftarrow \mathbb{P}_1, \ldots, i_k \leftarrow \mathbb{P}_k$$

The model M is evaluated n times, once for each vector of input variable samples. As a result, a data set of n evaluations of the model is obtained:

$$x_1 = M(v_1), \ldots, x_n = M(v_n)$$

Interpreting the Output. To obtain a final result of the model simulations, the output of the model simulations can be analyzed by statistics, such as the sample mean, the sample standard deviation, or further analysis techniques. The sample mean is defined as follows:

$$\overline{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i$$

The sample variance is defined as follows:

$$s_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2$$

Determining the Number of Monte-Carlo Simulations. When applying Monte-Carlo simulations to numerically address complex stochastic problems, one natural question arises: How many independent Monte-Carlo simulations should be executed? The answer to this question depends on several things as the following reasoning shows.

Let $x_1, \ldots, x_n$ be a sequence of samples of X, where X follows a distribution with statistical mean μ_X and statistical variance σ^2. Let $\overline{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i$ be a sample mean of X and $s_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x}_n)^2$ be the sample variance of X.

Assume a number of independent Monte-Carlo simulations (each with n runs) of the model are run. We obtain a random variable $\overline{X}$ for the distribution of the sample means of the model. It can be shown [DS04] that the statistical mean $\mu_{\overline{X}}$ of $\overline{X}$ corresponds to the statistical mean μ_X of X. Furthermore, the statistical variance of $\overline{X}$ relates to the statistical variance of X as $\sigma_{\overline{X}}^2 = \frac{\sigma_X^2}{n}$.

By the application of the Law of Large Numbers and the Central Limit Theorem, from the number of independent random variables n, a sample mean $\overline{x}_n$, and a sample variance s_n^2, the following confidence interval can be computed:

$$\left[\overline{x}_n - z \frac{s_n}{\sqrt{n}}, \overline{x}_n + z \frac{s_n}{\sqrt{n}} \right]$$

Confidence level	99.75%	99%	95.5%	95%	90%	68%	50%
z-score	3	2.58	2	1.96	1.65	1	0.67

Table 3.1: Relation between confidence levels of normal distributions and z-scores.

In the confidence interval, the value z is referred to as z-score and indicates the target confidence level. Given the confidence interval, one can conclude that the statistical mean of $\overline{X}$, respectively X, lies within the resulting confidence interval with a certain confidence level (expressed as z-score), *e.g.* a z-score of 2 indicates that the statistical means lies within the given confidence interval with a confidence of 95.5%. A relation between confidence levels and z-scores is provided in Table 3.1.

It can be concluded that the required number of independent Monte-Carlo simulations depends on the confidence (in terms of confidence level and confidence interval size) that one tries to achieve when approximating the statistical mean of a random variable by a sample mean of that random variable.

The application of Monte-Carlo simulations for risk estimations in the security context is not new. Noel *et al.* [NJWS10] introduce model building upon attack graphs with uncertain input variables. On the foundation of Monte-Carlo simulation, the model is evaluated to determine the probability of successful attacks against computer networks. While being general in description, the fundamental idea of their work builds the foundation for our contribution.

3.3.4. Pareto Dominance

In many disciplines it is necessary to select the most appropriate solution from a set of competing alternative solutions considering a set of (conflicting) decision criteria; these problems are generally referred to as *multi-criteria decision analysis* [Che06].

Let a finite set of possible solutions $X = \{x_1, \ldots, x_m\}$ and a set of decision criteria $\{c_1, \ldots, c_n\}$ be given. Let a set of objective functions $f_1 : X \to S_{c_1}, \ldots, f_n : X \to S_{c_n}$ be given with $S_{c_1}, \ldots, S_{c_n}$ being the performance scales of decision criteria $c_1, \ldots, c_n$. Given the fact that generally there is no solution that outperforms all other solutions with regard to all decision criteria, the decision process can be supported by identifying and discarding solutions that are dominated by other solutions. Pareto dominance, named after the Italian economist Vilfredo Pareto, denotes the fact that a solution performs worse than another solution with regard to all decision criteria. Formally, this can be defined as follows:

Definition 1 (Pareto Dominance)**.** *A solution $x \in X$ is Pareto dominated with regard to objective functions $f_1(x), \ldots, f_n(x)$ iff there is another solution $x' \in X$, such that $f_i(x') \geq f_i(x)$ for all i with $1 \leq i \leq n$ and $f_j(x') > f_j(x)$ for at least one j with $1 \leq j \leq n$.*

All solutions that are not Pareto dominated are *Pareto optimal*. The set of Pareto optimal points is called the *Pareto front*.

3.4. Building Blocks and Processes of the Framework

After the preliminaries of the security evaluation framework have been presented, this section is dedicated to its actual construction. Recall that the framework shall allow the election official to quantitatively measure the satisfaction degrees of Internet voting schemes with regard to legally-founded security requirements in an election-specific manner. An overview about the envisioned quantitative security evaluation is provided in Figure 3.3. In the first part of this section, a simple Internet voting scheme is presented (Section 3.4.1). The scheme serves to explicate the conceptual underpinnings of the security evaluation framework. After the legally-founded security requirements have been determined (refer to Sections 2.4 and 3.1.2), the block *Specification Languages Foundation* (Section 3.4.2) is dedicated to determining uniform adversarial capabilities. The block *Qualitative Security Models* (Section 3.4.3) provides a specification language to system analysts which they use to capture the qualitative security of Internet voting schemes. The resulting qualitative security models indicate which type of adversary can cause which impact to the different security requirements. The block *Election Setting* (Section 3.4.4) provides a specification language to election officials which they use to capture the expected adversary, among others. The block *Satisfaction Degree Determination Algorithm* (Section 3.4.5) defines an algorithm that evaluates the qualitative security models of an Internet voting scheme within the specified election setting. The output of this algorithm are satisfaction degrees for all security requirements.

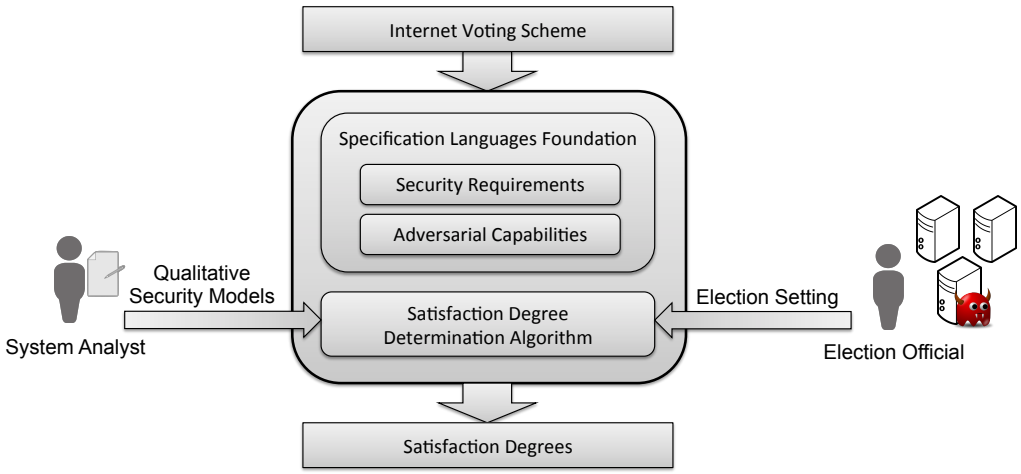

Figure 3.3: The security evaluation framework and its building blocks.

3.4.1. Exemplary Internet Voting Scheme

We introduce a simple Internet voting scheme to explain the concepts of the constructed security evaluation framework.

Components

We outline the components involved in the election and their respective roles.

Registration Server (RS). The registration server in collaboration with the validation server conducts eligibility checks and provides voters with voting tokens.

Validation Server (VS). The validation server in collaboration with the registration server conducts eligibility checks, generates voting tokens, and provides voters with voting tokens.

Ballot Box Server (BBS). The ballot box server provides eligible voters with the digital ballot, stores their filled ballots, and eventually calculates the election result.

Voting Device (VD). Each voter has a voting device at her disposal, which she uses to fill and cast her digital ballot.

Protocol Description

We describe the protocol underlying the example Internet voting scheme. The sequence diagram of the toy example is provided in Figure 3.4. We will provide cryptographic foundations of Internet voting schemes in Chapter 4 of this work. For the sake of clarity, within the example scheme, we assume that channels between the voting device and service providers and between service providers are authentic and confidential. In addition to that, no cryptographic techniques are in place, *i.e.* data is not encrypted.

Setup Phase. In advance to the election, credentials for all eligible voters are generated. These credentials are subsequently embedded into *RS* and *VS*. Furthermore, the digital ballot is embedded into *BBS*.

Voting Phase. To initiate the voting process on her voting device, a voter establishes a connection towards *RS* and authenticates herself towards *RS*. *RS* verifies the voter's eligibility and additionally consults *VS*. *VS* verifies the voter's eligibility one more time and generates a credential upon successful eligibility check. *VS* forwards that credential to *BBS* and *RS*, which in turn forwards the credential to the voter. In order to cast her vote, the voter consults the election website, hosted by *BBS*. She subsequently casts her vote together with her credential. *BBS* verifies the validity of the cast vote by checking whether the credential has been generated by *VS* and has not yet been used to cast a vote. Upon success, *BBS* stores the vote for the later vote tallying.

Tallying Phase. After all votes have been cast, *BBS* sums up all received votes and announces the election result.

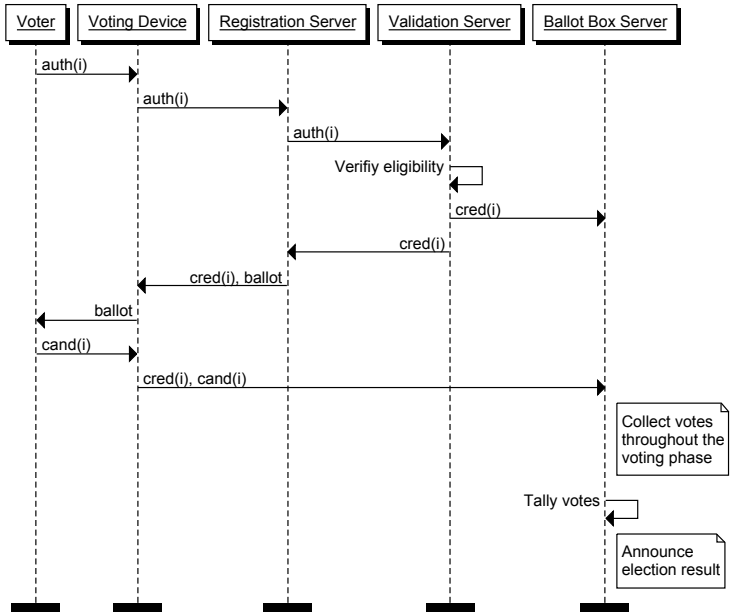

Figure 3.4: Sequence diagram of the example scheme.

3.4.2. Adversarial Capabilities and Specification of Qualitative Adversary Models

The foundation of the security evaluation is the specification of adversaries. While this specification must on the one side precisely describe successful adversaries against security requirements within Internet voting schemes, they must at the same time be sufficiently abstract to serve election officials to specify their election settings. We start by determining adversarial capabilities and show how they are composed to adversary models.

Adversarial Capabilities

We specify adversaries by a capability-based approach. In the capability-based approach, a mapping between security requirements and assumptions (exclusion of adversaries) under which those requirements can be ensured, is established.

We thereby follow the approaches by Langer [Lan10] and Carlos *et al.* [CMPC13]. Langer builds upon the well-established Dolev-Yao adversary model [DY83]. Langer adjusts the model with regard to a number of capabilities. A further extension of the Dolev-Yao adversary model has been proposed by Carlos *et al.* [CMPC13]. Starting from the concept of security ceremonies [Ell07] as extension of security protocols by human peers, Carlos *et al.* propose the ceremony- and context-dependent propagation of Dolev-Yao capabilities to human-human and human-device channels. Carlos *et al.*'s model has recently been applied to analyze the Helios voting scheme [MdSO+15].

As composition of previous research results, we classify adversarial capabilities in three

sub-classes, namely *corruption capabilities*, *channel capabilities*, and *computational capabilities*. Note that in the following paragraphs, variables are indicated by [*`X`*].

Corruption Capabilities. The security of Internet voting might be threatened by corrupt service providers carrying the election duties, be it either in terms of administrators, hardware, or software components. We distinguish between service providers that are not in direct contact with voters (*offline service providers*) and those that are in direct contact with voters (*online service providers*). It shall be emphasized that offline service providers are not necessarily disconnected from the Internet. We propose this distinction because of the difference in attack strategies required to compromise these service providers. While online service providers are generally threatened by external entities, such as malicious voters or hackers, the compromise of offline service providers in the most general case requires the collaboration of malicious insiders.

OFSP: The adversary can corrupt a [*`offline service provider`*].
ONSP: The adversary can corrupt an [*`online service provider`*].

On the voter-side, another crucial component is the device used to cast a vote. This device might be under adversarial control. Because of the fact that voters' device are generally used for a variety of purposes, controlling the voting device allows an adversary to learn the voter's identity.

VD: The adversary can corrupt a [*`voting device`*].

The security of the voting schemes does, however, not only depend on the trustworthiness of certain service providers or devices. Rather, these schemes' security relies on the human-computer interaction. Voters might be interested in or coerced into deviating from their original voting intention. We distinguish between the capabilities that the adversary might receive objects or data from voters (*voter output*), *e.g.* vote receipts, and that the adversary might provide voters with objects or data (*voter input*), *e.g.* instructions to cast a vote in a unique and identifiable manner.

VO: The adversary can receive objects/data from a [*`voter`*].
VI: The adversary can send objects/data to a [*`voter`*].

Channel Capabilities. We define one capability which indicates whether the adversary is capable of controlling the communication between voting devices and service providers or the communication between service providers.

CCH: The adversary can control a [*`communication channel`*] between a voting device and a service provider or between two service providers.

Possessing the capability *CCH* does not allow an adversary to determine the identity of message senders. We make the assumption that anonymization networks, *e.g. TOR*, are widely deployed and used by voters[7]. Carlos *et al.* [CMPC13] propose a refined Dolev-Yao model incorporating human-device communication channels, addressing so-called security ceremonies. However, Carlos *et al.* argue that assuming these new communication channels to be completely public might be too pessimistic. We follow that argumentation and define one adversarial capability indicating that the channel between a voter and her device(s) might be controlled by the adversary.

HCH: The adversary can control a [*`communication channel`*] between a voter and her voting device(s).

Computational Capabilities. A number or scientific works consider adversaries capable of obtaining (practically) unlimited computational resources, *e.g.* [MN06]. This is captured by the following capability.

CR: The adversary is computationally unrestricted.

In the remainder of this work, we will refer to the set

$$C = \{OFSP, ONSP, VO, VI, VD, CCH, HCH, CR\}$$

as *abstract capabilities* and to the (possibly infinite) set

$$C^A = \{OFSP_1, \ldots, OFSP_{n_1}, \ldots, HCH_1, \ldots, HCH_{n_7}, CR\}$$

as *instantiated capabilities* of scheme A if A captures n_1 offline service providers, ..., and n_7 voters.

Specification of Qualitative Adversary Models

The foundation of adversaries against Internet voting schemes are adversarial capabilities. However, adversaries must generally possess a number of these capabilities to violate security requirements, *e.g.* several service providers must be compromised. We define a *qualitative adversary model* as follows:

Definition 2 (Qualitative Adversary Model)**.** *Let an Internet voting scheme A with the set of instantiated capabilities C^A be given. A qualitative adversary model $\mathcal{A}_i^A$, or simply adversary, against scheme A is defined by a subset of instantiated capabilities C^A,* i.e. $\mathcal{A}_i^A \subseteq C^A$.

[7] It shall be emphasized that anonymization networks do generally have exit nodes that are capable of eavesdropping the communication. Consequently, in spite of the application of anonymization networks, the confidentiality of the communication must be ensured by other means.

In the remainder of this work, we consider a static adversary model, *i.e.* an adversary either does or does not possess capabilities throughout the entire election.

Example Scheme. For the sake of clarity, consider for the moment only the capabilities $ONSP$ and $OFSP$. Because RS and BBS are accessible by anybody, these components are online service providers. VS, in contrast, is not accessible to anybody and is therefore an offline service provider. To compromise RS and BBS, the adversary consequently needs the capabilities $ONSP_{RS}$ and $ONSP_{BBS}$. To compromise VS, the adversary needs the capability $OFSP_{VS}$. Hence, within the given Internet voting scheme, any set of capabilities $\mathcal{A}_i^A$ such that $\mathcal{A}_i^A \subseteq \{ONSP_{RS}, ONSP_{BBS}, OFSP_{VS}\}$ is a qualitative adversary model against the given scheme A.

3.4.3. Language for the Specification of Qualitative Security Models

In analogy to the definition of qualitative adversary models, we define qualitative security models within this subsection. Therefore, we first introduce the minimal cut sets notation. Thereafter, we discuss the impact that adversaries can cause on security requirements and ultimately define qualitative security models of Internet voting schemes.

Minimal Cut Sets

To specify qualitative security models, we pick up the concept of *minimal cut sets* [LGTL85]. Cut sets are a standard concept in reliability and availability theory [Ave85, IW89, ABdO76]. Fuqua [Fuq87] precisely describes cut sets as "*[...] any basic event or combination of basic events whose occurrence will cause the top event to occur.*" A cut set is *minimal*, if none of its subsets is a cut set. A violation of a security requirement (refer to Section 3.1.2) is the top event, while the possession of instantiated adversarial capabilities (refer to Section 3.4.2) corresponds to basic events. We call an adversary *successful* if the targeted security requirement can be violated with that adversary's capabilities.

Example Scheme. We stick to the restriction and consider only the capabilities $ONSP$ and $OFSP$. Throughout the voting phase, RS and VS learn the relation between the voter's identity and her voting credential. Furthermore, BBS learns the relation between voting credentials and the votes cast with those credentials. Hence, the malicious collaboration between RS and BBS or between VS and BBS results in a violation of vote secrecy. None of these servers might, however, violate vote secrecy individually. Consequently, the sets

$$\{ONSP_{RS}, ONSP_{BBS}\}, \{OFSP_{VS}, ONSP_{BBS}\}, \{ONSP_{RS}, OFSP_{VS}, ONSP_{BBS}\}$$

are cut sets. However, only the sets

$$\{ONSP_{RS}, ONSP_{BBS}\}, \{OFSP_{VS}, ONSP_{BBS}\}$$

are minimal as both sets are a subset of the cut set

$$\{ONSP_{RS}, OFSP_{VS}, ONSP_{BBS}\}.$$

Notational Conventions. For the sake of better readability, we introduce two notational conventions: First, as part of the scheme description, we make clear which capability is needed to compromise/control/influence which part of the scheme. Therefore, we omit the capability, but rather provide the component. For instance, instead of $ONSP_{RS}$ and $OFSP_{BBS}$, we simply write *RS* and *BBS*. Second, we rewrite lists of minimal cut sets in *disjunctive normal form*. Hence, rather than writing

$$\{RS, BBS\}, \{VS, BBS\},$$

we write

$$(RS \wedge BBS) \vee (VS \wedge BBS).$$

Adversarial Impact on Security Requirements

The violation of security requirements cannot be related to the presence of a unique adversary, but in fact different adversaries might cause different impact to security requirements. Consider for instance the security requirement vote integrity. Rather than assigning one precise successful adversary to that requirement, it is intuitive to indicate which adversaries might undetectably alter *one*, *two*, ..., or *all* cast votes. For example, an adversary controlling one voting device can manipulate one voter's vote, while an adversary controlling ten voting devices can manipulate ten voters' votes; an adversary controlling the ballot box server can even be in possession of an attack strategy manipulating all cast votes. Furthermore, there might be attacks that work up to a certain impact level, but could not cause the maximum impact. One typical attack of this form is a clash attack [KTV12], because clash attacks can only target two votes that are equal while their equality cannot be known in advance[8].

Hence, for each specific impact level, a qualitative security model is specified. Note that the number of impact levels depends on the number of eligible voters n_{el} and the number of expected voters n_{ex} and can therefore not be known throughout the determination of qualitative security models. Hence, qualitative security models are specified in a generic manner. Generally, attack strategies are successful up to a certain extent. For instance, the corruption of central servers would often result in the violation of a security requirement for all expected voters. In that case, the respective attack strategies are incorporated into all instantiated security models up to impact level n_{ex}. The corruption of one voting device might generally only violate a security requirement for one voter. Hence, once the numbers n_{el} and n_{ex} are known, the impact levels can be instantiated and so can the abstract qualitative security models.

[8]Knowing in advance to the election which voters will cast identical votes is at least very hard.

With regard to different security requirements, the impact on these requirements slightly differs. Vote secrecy, vote integrity, and fairness are only defined for voters that actually cast a vote. Hence the maximum impact of these requirements is n_{ex}. Eligibility relates to those voters that abstain from the election. Hence, an adversary causes maximum impact on eligibility if he is able to cast illegitimate votes for all $n_{el} - n_{ex}$ abstaining voters. Ultimately, with regard to data access protection, an adversary might be interested in obtaining voter data of all eligible voters. Hence, an adversary causes maximum impact on data access protection if he is able to obtain voter data for all n_{el} eligible voters. In the remainder of this work, we denote the maximum impact generically by n, an abstraction of n_{ex}, $n_{el} - n_{ex}$, and n_{el} respectively.

Definition of Qualitative Security Models

After the definition of qualitative adversary models and the discussion of adversarial impact of security requirements, we are able to define qualitative security models.

Definition 3 (Qualitative Security Model)**.** *Let an Internet voting scheme A with the set of instantiated capabilities C^A be given. We say that*

$$\mathcal{M}^{A,r,i} = (\alpha_1^{A,r,i} \vee \cdots \vee \alpha_{\xi^{A,r,i}}^{A,r,i})$$
$$\text{with } \alpha_j^{A,r,i} = (c_{j,1}^{A,r,i} \wedge \cdots \wedge c_{j,\lambda_j^{A,r,i}}^{A,r,i}) \text{ and } c_{j,k}^{A,r,i} \in C^A$$

is a qualitative security model of A with regard to security requirement r and impact level i if there exists a set of adversaries $\mathcal{S} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{\xi^{A,r,i}}\}$ where $\mathcal{A}_j$ is specified by capabilities $\{c_{j,1}^{A,r,i}, \ldots, c_{j,\lambda_j^{A,r,i}}^{A,r,i}\}$, such that

1. *The capabilities of all adversaries $\mathcal{A} \in \mathcal{S}$ suffice to cause impact i on r, and*
2. *For all adversaries $\mathcal{A} \in \mathcal{S}$, there is no adversary $\mathcal{A}' \subset \mathcal{A}$ such that the capabilities of $\mathcal{A}'$ suffice to cause impact i on r, and*
3. *For all adversaries $\mathcal{A}'$, of which the capabilities suffice to cause impact i on r, there is an adversary $\mathcal{A} \in \mathcal{S}$, such that $\mathcal{A} \subseteq \mathcal{A}'$.*

Intuitively speaking, a qualitative security model encodes a number of successful attack strategies (disjunctions), where each attack strategy requires the adversary to possess a number of instantiated capabilities (conjunctions). Here, it shall be emphasized that different attack strategies might overlap, *i.e.* different minimal cut sets $\alpha_j^{A,r,i}$ and $\alpha_x^{A,r,i}$ might contain identical instantiated capabilities. Hence, two capabilities $c_{j,k}^{A,r,i}$ and $c_{x,y}^{A,r,i}$ with $j \neq x$ might be identical.

Definition 4 (Resistance Against Abstract Capability)**.** *Let an Internet voting scheme A with the set of instantiated capabilities C^A and the qualitative security models $\mathcal{M}^{A,r,1}, \ldots, \mathcal{M}^{A,r,n}$*

be given. We say that the scheme A is resistant against capability $C_o \in C$ with regard to requirement r, if for all impact levels i, it holds that for all $c_{j,k}^{A,r,i}$ in all $\alpha_j^{A,r,i}$, $c_{j,k}^{A,r,i}$ is no instantiation of C_o.

Example Scheme: We relax the previous restriction and, in addition to the capabilities $OFSP$ and $ONSP$, consider that an adversary might gain the capability VD (corruption of voting devices). Hence, to cause impact 1 the adversary might either compromise any single voting device, or any two voting devices, or ... or all voting devices. Furthermore the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. To cause impact 2, the adversary might either compromise any two voting devices, or any three voting devices, or ... or all voting devices. Furthermore the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. Generically, to cause impact i, the adversary might either compromise any i voting devices, or any $i+1$ voting devices, or ... or all voting devices. Furthermore, the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. The resulting qualitative security models of the example scheme are provided in Table 3.2.

Requirement	Qualitative Security Models	Impact
Vote Secrecy	$(VD_1 \vee VD_2 \vee \cdots \vee VD_n) \vee$	
	$((VD_1 \wedge VD_2) \vee (VD_1 \wedge VD_3) \vee \cdots \vee (VD_{n-1} \wedge VD_n)) \vee$	
	$(RS \wedge BBS) \vee (VS \wedge BBS)$	1
	...	...
	$\bigvee_{I \subseteq \{1,\ldots,n\}, \lvert I \rvert \geq l} (\bigwedge_{i \in I} VD_i) \vee (RS \wedge BBS) \vee (VS \wedge BBS)$	$1 \leq l \leq n$

Table 3.2: Qualitative security models of the example scheme for vote secrecy.

Pareto Dominance of Internet Voting Schemes

Internet voting schemes can be *partially ordered* over qualitative security models. This is for instance of relevance when system developers target at improving an Internet voting scheme in one dimension, while not negatively affecting the scheme in any other dimension or when security analysts want to discard Internet voting schemes from further consideration without negatively affecting the quality of any election officials' decision.

Recall the definitions of qualitative adversary models (Definition 2) and qualitative security models (Definition 3). We propose the following definitions:

Definition 5 (Mapping of Scheme Capabilities)**.** *Let two Internet voting schemes A and B, and their respective instantiated capabilities C^A and C^B be given. We say that ϕ :*

$C^A \rightarrow C^B$ is a mapping of scheme capabilities iff ϕ is a bijection and for all $c \in C^A$, capabilities c and $\phi(c)$ are instantiations of the same abstract capability.

Definition 6 (Equality/Dominance of Qualitative Security Models)**.** *Let two Internet voting schemes A and B, and their respective qualitative security models $\mathcal{M}^{A,r,l}$ and $\mathcal{M}^{B,r,l}$ for all security requirements $r \in R$ and all impacts levels l be given. Furthermore assume a mapping ϕ of scheme capabilities of both schemes is given. We say that scheme A* equals *(=-relation) /* dominates *(>-relation) scheme B with regard to requirement r, impact level l, and mapping ϕ, if the following holds:*

1. *For each adversary $\mathcal{A}_i^A \subseteq C^A$ that satisfies $\mathcal{M}^{A,r,l}$, the adversary $\phi(\mathcal{A}_i^A)$ also satisfies $\mathcal{M}^{B,r,l}$, and*

2. a) *for each adversary $\mathcal{A}_j^B \subseteq C^B$ that satisfies $\mathcal{M}^{B,r,l}$, the adversary $\phi^{-1}(\mathcal{A}_j^B)$ also satisfies $\mathcal{M}^{A,r,l}$. (=-relation)*

   b) *not for each adversary $\mathcal{A}_j^B \subseteq C^B$ that satisfies $\mathcal{M}^{B,r,l}$, the adversary $\phi^{-1}(\mathcal{A}_j^B)$ also satisfies $\mathcal{M}^{A,r,l}$. (>-relation)*

Definition 7 (Pareto Dominance of Internet Voting Schemes)**.** *Let two Internet voting schemes A and B, and their respective qualitative security models $\mathcal{M}^{A,r,l}$ and $\mathcal{M}^{B,r,l}$ for all security requirements $r \in R$ and all impacts levels l be given. We say that scheme A* Pareto dominates *scheme B with regard R if there is a mapping ϕ of capabilities of both schemes such that:*

1. *for each requirement $r \in R$ and each impact level l, scheme A equals or dominates scheme B with regard to requirement r impact level l, and mapping ϕ, and*

2. *there is at least one requirement $r \in R$ and impact level l such that scheme A dominates scheme B with regard to requirement r impact level l, and mapping ϕ.*

The fundamental idea of this dominance is the following: If a scheme A Pareto dominates a scheme B with regard to all security requirements for Internet voting schemes (refer to Section 3.1.2), then the quantitative security evaluation results of scheme A shall be equal or better than the quantitative security evaluation results of scheme B with regard to all security requirements. Hence, from a security perspective, quantitative security evaluation becomes obsolete when determining the most appropriate scheme from these two schemes. Therefore, the notion of Pareto dominance serves system developers to improve Internet voting schemes in an election-independent manner.

3.4.4. Language for the Specification of Election Settings

If Internet voting schemes cannot be ordered in the sense of Definition 7, then quantitative election-specific security evaluation of the schemes is necessary. Adversarial capabilities are not only the underpinning of qualitative security models, but furthermore they form

the basis for the specification of election settings. The goal of this section is the definition of a specification language for election officials.

The quantitative evaluation of qualitative security models could be conducted in a simple manner if election officials could precisely assign probabilities to the presence of adversarial capabilities (refer to Section 3.4.2). However, election officials might provide these probabilities with some uncertainty due to the lack of available knowledge regarding capabilities. Furthermore, because of the potential complexity of qualitative security models, their quantitative evaluation might be significantly impacted by minor changes in capability probabilities. We take account of this and incorporate Monte-Carlo simulations into the quantification process. Rather than precise capability probabilities, we require the election official to provide probability distributions for abstract adversarial capability probabilities.

Additionally, the election official specifies the number of *eligible voters* n_{el} and estimates the *number of expected voters* n_{ex}. These numbers are needed to instantiate all possible impact levels. Eventually, election settings are defined as follows:

Definition 8 (Election Setting)**.** *Given the set of abstract capabilities C, the number of eligible voters n_{el}, the number of expected voters n_{ex}, and probability distributions $\mathbb{P}_{C_o}$ for all capabilities $C_o \in C$, we say that the tuple*

$$E = (\mathbb{P}_{C_1}, \ldots, \mathbb{P}_{C_{|C|}}, n_{el}, n_{ex})$$

is an election setting.

Example Election Setting. We consider the following election setting, where $U[a, b]$ denotes the uniform distribution with support (a, b):

$$\begin{aligned}
E = (&\mathbb{P}_{OFSP} = U[0.0001, 0.0005]; && \text{Distribution for capability OFSP}\\
&\mathbb{P}_{ONSP} = U[0.001, 0.005]; && \text{Distribution for capability ONSP}\\
&\mathbb{P}_{VO} = U[0.01, 0.05]; && \text{Distribution for capability VO}\\
&\mathbb{P}_{VI} = U[0.01, 0.05]; && \text{Distribution for capability VI}\\
&\mathbb{P}_{VD} = U[0.01, 0.05]; && \text{Distribution for capability VD}\\
&\mathbb{P}_{CCH} = U[1, 1]; && \text{Distribution for capability CCH}\\
&\mathbb{P}_{HCH} = U[0.01, 0.05]; && \text{Distribution for capability HCH}\\
&\mathbb{P}_{CR} = U[0, 0]; && \text{Distribution for capability CR}\\
&n_{el} = 2,000; && \text{Number of eligible voters}\\
&n_{ex} = 1,000) && \text{Number of expected voters}
\end{aligned}$$

3.4.5. Determination of Satisfaction Degrees in Election Settings

The core of the framework is the algorithm for the quantitative evaluation of qualitative security models within specific election settings. Therefore, it is first shown how the probability of an adversary violating a qualitative security model can be calculated. Thereafter, it is outlined how satisfaction degrees can be calculated with given fixed probabilities for adversarial capabilities. Eventually, we show how Monte-Carlo simulations [MU49] are adapted for the quantitative evaluation of qualitative security models against probabilistic adversaries.

We abbreviate the probability of the event that the adversary $\mathcal{A}$ satisfies a security model X or possesses a specific (abstract or instantiated) capability, *i.e.* $P_{\mathcal{A}}(X = 1)$, by $P(X)$.

Transforming Qualitative Security Models into Probability Formulas

The probability distribution $\mathbb{P}_{C_o} : [0,1] \to [0,1]$ for an abstract capability $C_o \in C$ has as events probabilities $P(C_o)$. Suppose now that for each $C_o \in C$ there is such an event $P(C_o)$ given. All instantiated capabilities of C_o inherit the probabilities $P(C_o)$ and are independent from each other. Hence, for any two instantiations $c_{j,k}^{A,r,i}$ and $c_{x,y}^{A,r,i}$ of the same abstract capability C_o, it holds $P(c_{j,k}^{A,r,i}) = P(c_{x,y}^{A,r,i}) = P(C_o)$. Then one can compute the probability that an adversary satisfies $\alpha_j^{A,r,i}$ as:

$$P(\alpha_j^{A,r,i}) = P(c_{j,1}^{A,r,i}) \cdot P(c_{j,2}^{A,r,i}) \cdot \dots \cdot P(c_{j,\lambda_j^{A,r,i}}^{A,r,i})$$

Ultimately, we are interested in the probability that an adversary might cause impact i on requirement r in scheme A, *i.e.* the probability $P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i})$. The *inclusion-exclusion principle* [KLS96, Vau98] provides a means to calculate the probability that at least one of several (possibly overlapping) events happens. Consequently, to calculate the probability $P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i})$, the application of the inclusion-exclusion principle leads to the following probability:

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}) = \sum_{j=1}^{\xi^{A,r,i}} \left((-1)^{j-1} \sum_{J \subset \{1,\dots,\xi^{A,r,i}\}, |J|=j} P(\bigwedge_{b \in J} \alpha_b^{A,r,i}) \right) \tag{3.1}$$

If none of the minimal cut sets overlap, then one can apply De Morgan's Law. Hence, the resulting probability formulas for non-overlapping minimal cut sets is:

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}) = 1 - ((1 - P(\alpha_1^{A,r,i})) \cdot (1 - P(\alpha_2^{A,r,i})) \cdot \dots \cdot (1 - P(\alpha_{\xi^{A,r,i}}^{A,r,i})))$$

In addition to the inclusion-exclusion principle, to calculate the probability of an adversary satisfying a minimal cut set of the form "*at least* x events", instances of the same abstract

capability (*e.g.* in the case of voting device corruption), the *cumulative binomial probability* computation is applied. Finally, the resulting probability formulas build the foundation for quantitative security evaluation. Note that system analysts might provide probability formulas directly rather than qualitative security models. However, the transformation of qualitative security models would require the system analyst to consider the overlappings of different attack strategies and the mathematical modelling of those overlappings. To lower the system analyst's burden, the transformation is incorporated into the framework's quantification process.

Example Scheme. We first consider the probability of the event that either the registration server *and* the ballot box server *or* the validation server *and* the ballot box server are compromised (event A). Therefore, we apply the inclusion-exclusion principle:

$$P(A) = P(RS) \cdot P(BBS) + P(VS) \cdot P(BBS) - P(RS) \cdot P(VS) \cdot P(BBS)$$

Furthermore, we consider the probability of the event that at least l voting devices are compromised (event B):

$$P(B) = 1 - \left(\sum_{i=0}^{l-1} \binom{n}{i} P(VD)^i \cdot (1 - P(VD)^{n-i}) \right)$$

Given the independence of events A and B (no overlappings), we can compute the probability that vote secrecy of at least l votes is violated as follows:

$$P(A \cup B) = 1 - ((1 - P(A)) \cdot (1 - P(B)))$$

Determination of Satisfaction Degrees with Given Probabilities

The evaluation of qualitative security models within election settings is built upon standard risk theory (refer for instance to [SGF02]). To determine the satisfaction degree of an Internet voting scheme A with qualitative security models $\mathcal{M}^{A,r,i}$ under given probabilities $P(C_o)$ for all $C_o \in C$ and under n impact levels (the instantiation of impact levels will be explained in the following paragraph), the following function $f(P(C_1), \ldots, P(C_{|C|}))$ is defined:

1. For each instantiated impact level $1 \leq i \leq n$, the probability formula of the qualitative security model is evaluated based on the given probabilities.
2. For each instantiated impact level $1 \leq i \leq n$, a risk value is calculated by multiplying the normalized impact $\frac{i}{n}$ with the evaluated probability formula of the respective qualitative security model.
3. The largest risk value is identified.
4. The satisfaction degree estimator is the inverse of the largest risk value.

Extension towards Probabilistic Adversaries

Recall that election officials assign probability distributions rather than precise probabilities to adversarial capabilities. Following the Monte-Carlo approach, the given distributions (refer to Section 3.4.4) are sampled and the qualitative security models are evaluated with those random samples. To determine the satisfaction degree of an Internet voting scheme A with qualitative security models $\mathcal{M}^{A,r,i}$ with regard to a security requirement $r \in R$ (refer to Section 2) within a specified election setting $E = (\mathbb{P}_{OFSP}, \mathbb{P}_{ONSP}, \mathbb{P}_{VO}, \mathbb{P}_{VI}, \mathbb{P}_{VD}, \mathbb{P}_{CCH}, \mathbb{P}_{HCH}, \mathbb{P}_{CR}, \mathrm{n}_{el}, n_{ex})$ (refer to Section 3.4.4), the following process is defined:

Instantiation of Impact Levels. Based on the number of eligible voters n_{el} and the number of expected voters n_{ex} (refer to Section 3.4.4), the number of impact levels is instantiated and probability formulas of qualitative security models accordingly. Consequently, n (depending on the security requirement under investigation either n_{ex}, $n_{el} - n_{ex}$, or n_{el}) impact levels are assigned to n probability formulas. The probability formula for causing impact i against vote secrecy within the example scheme is given in Section 3.4.5.

Generation of Monte-Carlo based Satisfaction Degree Estimators. The following steps are conducted m times (number Monte-Carlo iterations). The process steps are shown for the j-th Monte-Carlo iteration.

1. For each abstract adversarial capability $C_o \in C$ (refer to Section 3.4.2), an estimator of the probability $P(C_o)$ is sampled according to the probability distribution $\mathbb{P}_{C_o}$ in E (refer to Section 3.4.4). For the example election setting this could lead to the following probability samples: $P^1(OFSP) = 0.000232$, $P^1(ONSP) = 0.004283$, $P^1(VO) = 0.02482$, $P^1(VI) = 0.03993$, $P^1(VD) = 0.04832$, $P^1(CCH) = 1$, $P^1(HCH) = 0.04813$, $P^1(CR) = 0$.

2. For the vector of probability samples, the deterministic satisfaction degree calculator f is called. The process steps are outlined in the following:

   a) For each instantiated impact level $1 \leq i \leq n$, the probability formula of the qualitative security model is evaluated based on the samples generated in step 1. We provide an excerpt of this step for the example scheme:

   b) For each instantiated impact level $1 \leq i \leq n$, a risk value is calculated by multiplying the normalized impact $\frac{i}{n}$ with the evaluated probability formula of the respective qualitative security model (result of step 2.a). We provide an excerpt of this step for the example scheme:

   c) The largest risk value (result of step 2.b) is identified. In the example scheme, the largest risk value appears at impact level 39 and equals 0.03534598941.

Impact	Probability (Qualitative Security Models)
1	1
⋮	⋮
39	0.9063074207
⋮	⋮
1000	0.00001933348919

Impact	Probability (Qualitative Security Models)	Risk
1	1	0.001
⋮	⋮	⋮
39	0.9063074207	0.03534598941
⋮	⋮	⋮
1000	0.00001933348919	0.00001933348919

d) The satisfaction degree estimator is the inverse of the largest risk value (result of step 2.c). The value is denoted by satisfaction degree estimator e_j in the j-th Monte-Carlo simulation. For the example scheme, the satisfaction degree estimator is 0.96465401059.

Conducting these two steps with random variables $P(C_1), \ldots, P(C_{|C|})$ yields samples of the following random variable:

$$M := f(P(C_1), P(C_2), \ldots, P(C_{|C|}))$$

Processing of Satisfaction Degree Estimators. We define the *statistical satisfaction degree* of scheme A with regard to requirement r and election setting E as the expected value of random variable M, *i.e.* $\mathbb{E}(M)$.

1. To approximate $\mathbb{E}(M)$ by the m generated satisfaction degree estimators, namely $e_1, \ldots, e_m$, the average of these estimators is calculated. Hence, the *empirical satisfaction degree* $\overline{M^m}$ (in the remainder simply referred to as satisfaction degree) of scheme A with regard to requirement r and election setting E is defined as:

$$\overline{M^m} := \frac{1}{m}(e_1 + \cdots + e_m) = \frac{1}{m}\sum_{k=1}^{m} f(P^k(C_1), P^k(C_2), \ldots, P^k(C_{|C|}))$$

By the weak law of large numbers, it holds that the empirical satisfaction degree weakly converges towards the statistical satisfaction degree,

$$\overline{M^m} \xrightarrow{m\to\infty} \mathbb{E}[M].$$

The satisfaction degree of the example scheme after 10,000 Monte-Carlo simulations equals 0.9786.

2. To evaluate the quality of the empirical satisfaction degree with regard to the statistical satisfaction degree, a confidence interval is calculated. Mathematically, the confidence interval surrounding the empirical satisfaction degree contains the statistical satisfaction degree with a certain confidence and is calculated as follows:

$$CI = \left[\overline{M^m} - z\frac{s_m}{\sqrt{m}}, \overline{M^m} + z\frac{s_m}{\sqrt{m}}\right]$$

The value z is referred to as *confidence value* and indicates the confidence with which the statistical satisfaction degree is within the calculated confidence interval. An overview of confidence values and the resulting confidence in percentage is for instance provided by Driels and Shin [DS04]. The value s_m denotes the standard deviation.

For the evaluation of the example scheme, we set the confidence value to $z = 2$, thereby obtaining a certainty of $\approx 95.5\%$ that the statistical mean lies within the confidence interval generated around the empirical mean. The confidence interval of the example scheme after 10,000 Monte-Carlo simulations is:

$$[0.9783, 0.9787]$$

In the following security evaluation, we make sure that confidence intervals of compared Internet voting schemes do not overlap unless both schemes build upon the same capabilities. As a consequence thereof, we omit confidence intervals in the remainder of this work.

It shall be emphasized that the herein presented construction composes risk values assigned to different impact levels in a restrictive way, *i.e.* the largest identified risk value serves as indicator for the computation of the satisfaction degree (see step 2.d). We justify this decision by the fact that adversaries have profound knowledge about the Internet voting scheme in use and will choose the most effective strategy to achieve their goals. This decision is, however, by no means set in stone. The construction might easily be adapted to incorporate a less restrictive model, *e.g.* averaging over all identified risks.

A Note on Monte-Carlo Simulations

As rule of thumb, one can say that the larger the number of Monte-Carlo iterations m and the smaller the confidence value z, the smaller the resulting confidence interval, and hence

the higher the robustness of the empirical satisfaction degree estimation. Consequently, for the application of Monte-Carlo simulations, the planned number of Monte-Carlo iterations m and the confidence value z, have to be specified. With regard to the number of Monte-Carlo simulations, we follow the recommendations by Mundform *et al.* [MSK+11] and set $m = 10,000$. Additionally, we set the confidence value to $z = 2$.

3.5. Deduction of Qualitative Security Models and Determination of Election Settings

After the security evaluation framework has been constructed, we provide brief guidelines about how system analysts might deduce qualitative security models for Internet voting schemes and how election officials might determine their election setting.

Deduction of Qualitative Security Models. The system analyst has a number of techniques at disposal to determine qualitative security models, such as symbolic protocol analysis, *e.g.* [BPM02], cryptographic proof techniques, *e.g.* [BR93], and threat analysis, *e.g.* [PYL10]. In fact, there is a recent tendency towards automating the deduction of adversarial capabilities upon which a protocol builds. So far, these approaches do, however, either consider a very limited class of cryptographic protocols [BC14] or are tailored towards specific security requirements [NV12]. Given the difference in their rigor, the used technique might correlate to the reliability of the output. We consequently recommend to provide qualitative security models together with the approach used to deduce these models.

Determination of Election Settings. Given the facts that election officials might be overwhelmed with assigning precise probabilities to adversarial capabilities and that the quantitative evaluation of qualitative security models might result in major changes under minor probability changes, the constructed framework allows election officials to assign probability distributions to adversarial capabilities.

There exist estimations regarding different adversarial capabilities. For instance, PandaLabs security provides quarterly security reports which contain infection rates of general-purpose machines. For instance, according to the July-September 2015 report [Pan15], China has an infection rate of $\approx 45\%$, Germany a rate of $\approx 25\%$, and Norway a rate of $\approx 20\%$. These values might be serve as indicator for infection rates of voter's devices[9]. Additionally, election officials might build their estimations upon past experience as proposed by Schryen *et al.* [SVRH11].

Depending on the available information, election settings might be defined in three ways: If information about specific adversarial capabilities is rare or the certainty about

[9]It should however be noted that only a fraction of infected voting devices would provide an adversary sufficient control to influence the election.

precise information is low, then probability distributions have a larger variance. If, on the other side, election officials have precise and certain information regarding the expected adversary, than probability distributions have a smaller variance. In fact, if available information regarding the election setting is very rare, then election official might determine the most appropriate scheme depending on a variety of (pre-defined) election settings.

3.6. Properties of the Security Evaluation Framework

After its construction, the security evaluation framework is evaluated with regard to the requirements determined in Section 3.3.1. The following proofs build upon the weak law of large numbers and hold therefore for a sufficiently large number of Monte-Carlo iterations.

3.6.1. No Capabilities – Perfect Security

The first requirement that the security evaluation framework shall possess is that the satisfaction degree of all schemes must be 1 with regard to all security requirements, if the adversary has no capabilities, unless the security requirement can be violated without any adversarial capabilities. This void of capabilities is equivalent to the absence of randomness as the adversary's capability is determined. Hence the probability distributions that are passed by the election official, degenerate to deterministic functions. Within a probabilistic framework, such deterministic functions are called constant random variables. Their distribution function is the *Dirac delta function* δ_x, where $x \in \mathbb{R}$ denotes the point of mass [Haz01]. In particular, it holds $U(a, a + 1/n) \xrightarrow{n\to\infty} \delta_a$. Hence, for each $C_o \in C$ the Dirac delta function δ_o is passed, as there is only one probability that can be assigned to the event that an adversary has capability C_o, namely *zero*.

Theorem 3.1. *Let δ_o be the distribution function for all abstract capabilities $C_o \in C$. The satisfaction degree of scheme A is 1 for all security requirements r, unless the security requirement can be violated without any adversarial capabilities.*

Proof. If the probability of having an abstract capability $C_o \in C$ is 0 for all $C_o \in C$, then all instantiated capabilities $c_{j,k}^{A,r,i}$, with $1 \leq l \leq \lambda_j^{A,r,i}$ for the impact level i have probability 0, i.e. $P(c_{j,k}^{A,r,i}) = 0$. This leads to $P(\alpha_j^{A,r,i}) = 0$ and thus

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}) \leq \sum_{j=1}^{\xi^{A,r,i}} P(\alpha_j^{A,r,i}) = 0.$$

As this holds true for all impact levels, the maximum risk of all impact levels equals 0. Consequently, the satisfaction degree estimator results in 1. Given the fact that the random variables for capability probabilities have their entire density at 0, each Monte-Carlo iteration assigns the value 0 to all capability probabilities. Hence, the resulting random variable M has its entire density on the value 1, such that $\mathbb{E}(M) = 1$. □

3.6.2. Capability Resistance

The second requirement refers to the resistance of Internet voting schemes against specific abstract adversarial capabilities.

Theorem 3.2. *Let Internet voting scheme A be resistant against abstract capability C_o with regard to requirement r. Let $P(C_1), \ldots, P(C_o), \ldots, P(C_{|C|})$ denote random variables for the probabilities of adversarial capabilities $C_1, \ldots C_o, \ldots C_{|C|}$. If random variable $P(C_o)$ is replaced by a differently distributed random variable $P(C_o)'$, then the resulting satisfaction degrees of scheme A with regard to requirement r do not differ.*

Proof. For the random variables $P(C_1), \ldots, P(C_o)', \ldots, P(C_{|C|})$, we denote the random variable generated by the Monte-Carlo simulations by:

$$M' := f(P(C_1), \ldots, P(C_o)', \ldots, P(C_{|C|}))$$

Due to A's resistance, it holds for all $c_{j,k}^{A,r,i}$ in all $\alpha_j^{A,r,i}$ that $c_{j,k}^{A,r,i}$ is no instantiation of C_o. Consequently, function f is neither affected by random variable $P(C_o)$ nor by $P(C_o)'$. As a consequence, it holds

$$\begin{aligned} M &= f(P(C_1), \ldots, P(C_o), \ldots, P(C_{|C|})) \\ &= f(P(C_1), \ldots, P(C_o)', \ldots, P(C_{|C|})) = M', \end{aligned}$$

and hence $\mathbb{E}(M) = \mathbb{E}(M')$. □

3.6.3. Continuity

Election officials provide uniform probability distributions for capability probabilities, *e.g.* distributions $P(C_i) \sim U[a_i, b_i], i = 1, 2, \ldots, |C|$. To prove continuity of the framework with regard to the expected adversary, we study the framework's result under sequences of random variables $(P(C_{i,n}))_{n \in \mathbb{N}}$ where $P(C_{i,n}) \sim U[a_i, b_i + 1/n]$ for $i = 1, 2, \ldots, |C|$. We say that continuity is given if the framework's results are identical under the random variables $P(C_i) \sim U[a_i, b_i]$ and $P(C_{i,n}) \sim U[a_i, b_i + 1/n]$ for n converging to infinity. Formally, this is expressed as follows:

$$\begin{aligned} \mathbb{E}(M_n) &= \mathbb{E}(\ f(P(C_{1,n}), P(C_{2,n}), \ldots, P(C_{|C|,n}))\ ) \\ &\overset{n \to \infty}{\longrightarrow} \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_{|C|}))\ ) = \mathbb{E}(M) \end{aligned}$$

Before proving the main theorem, we define and prove three lemmata.

Lemma 3.3. *The probability that an adversary causes impact i on requirement r in scheme A is continuous with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. The probability of $P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i})$ can be calculated by multiplying *once* the probabilities $P(c_j)$ for which c_j appears in either $\alpha_x^{A,r,i}$ or $\alpha_y^{A,r,i}$.

Suppose w.l.o.g. that the instantiated capabilities $c_{x,\lambda_x^{A,r,i}}^{A,r,i}$ and $c_{y,\lambda_y^{A,r,i}}^{A,r,i}$ are equal, hence $\alpha_x^{A,r,i}$ and $\alpha_y^{A,r,i}$ overlap. Then, the probability of $P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i})$ is calculated as:

$$P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i}) = P(c_{x,1}^{A,r,i}) \cdot \dots \cdot P(c_{x,\lambda_x^{A,r,i}}^{A,r,i}) \cdot P(c_{y,1}^{A,r,i}) \cdot \dots \cdot P(c_{y,(\lambda_y^{A,r,i}-1)}^{A,r,i})$$

On the other side, if $\alpha_x^{A,r,i}$ and $\alpha_y^{A,r,i}$ do not overlap, the probability of $P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i})$ is calculated as:

$$P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i}) = P(c_{x,1}^{A,r,i}) \cdot \dots \cdot P(c_{x,\lambda_x^{A,r,i}}^{A,r,i}) \cdot P(c_{y,1}^{A,r,i}) \cdot \dots \cdot P(c_{y,\lambda_y^{A,r,i}}^{A,r,i})$$

Consequently, for any intersection of (possibly overlapping) minimal cut sets $\alpha_x^{A,r,i}$ and $\alpha_y^{A,r,i}$, the probability of the intersection is given by a product of probabilities $P(c_{a,b}^{A,r,i})$. Given the fact that the product of continuous functions is again continuous [GJ13], the value $P(\alpha_x^{A,r,i} \wedge \alpha_y^{A,r,i})$ is continuous with regard to the probabilities $P(C_o)$ for $C_o \in C$.

Consider the event that at least one of several minimal cut sets (causing impact i on requirement r in scheme A) is satisfied by the adversary. The probability of the event that at least one of several minimal cut sets (causing impact i on requirement r in scheme A) is satisfied by the adversary, is given by Formula 3.1. The calculation of this probability builds upon the addition and subtraction of products; namely the products defined by $P(\bigwedge_{j \in J} \alpha_j^{A,r,i})$. Given the facts that the addition and subtraction of continuous functions is again continuous [GJ13] and that the products $P(\bigwedge_{j \in J} \alpha_j^{A,r,i})$ are continuous functions, the value

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i})$$

is continuous with regard to the probabilities $P(C_o)$ for $C_o \in C$. □

Lemma 3.4. *The satisfaction degree estimator for requirement r in scheme A is continuous with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. According to the security evaluation framework, the risk of requirement r in scheme A and impact level i can be calculated by:

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}) \cdot \frac{i}{n}$$

Given the fact that the product of continuous functions is again continuous [GJ13] and Lemma 3.3, the risk is continuous with regard to the probabilities $P(C_o)$ for $C_o \in C$.

Subsequently, the security evaluation framework determines the maximum of the n (all impact levels) computed risks with regard to requirement r as direct indicator for the satisfaction degree of the scheme with regard to r. It can be shown that the maximum of continuous functions is again continuous [Str00].

Ultimately, given the fact that the subtraction of continuous functions is again continuous [GJ13], the satisfaction degree estimator is continuous. □

Definition 9. *A sequence of random variables* $(X_n)_{n\in\mathbb{N}}$ weakly converges *to a random variable* X*, if for every continuous function* f*, it holds*

$$lim_{n\to\infty}\int_{X_n} f(x)\mathrm{d}\mathbb{P}_{X_n} = \int_X f(x)\mathrm{d}\mathbb{P}_X,$$

where $\mathbb{P}_{X_n}$ *denotes the probability distribution of* X_n *and* $\mathbb{P}_X$ *the probability distribution of* X*, shortly*

$$X_n \xrightarrow{d} X.$$

Lemma 3.5. *Let* $X \sim U[a,b]$ *be a uniformly distributed random variable and let* $(X_n)_{n\in\mathbb{N}} \sim U(a, b+1/n)$ *be a sequence of random variables. Then it holds*

$$X_n \xrightarrow{d} X.$$

Proof. We have for any continuous function f:

$$\begin{aligned} lim_{n\to\infty}\int_a^{b+1/n} \frac{1}{b+1/n-a} f(x)\mathrm{d}x &= lim_{n\to\infty}\frac{1}{b+1/n-a}\int_a^{b+1/n} f(x)\mathrm{d}x \\ &= lim_{n\to\infty}\frac{1}{b+1/n-a} lim_{n\to\infty}\int_a^{b+1/n} f(x)\mathrm{d}x \\ &= \frac{1}{b-a}\int_a^b f(x)\mathrm{d}x \\ &= \int_a^b \frac{1}{b-a} f(x)\mathrm{d}x \end{aligned}$$

□

Theorem 3.6. *Let* $P(C_i) \sim U[a_i, b_i], i = 1, 2, \ldots, |C|$ *denote uniformly distributed random variables for the probabilities of adversarial capabilities* C_i*. The satisfaction degree of* A *with regard to requirement* r *is continuous with regard to any weakly convergent sequence of random variables* $(P(C_{i,n}))_{n\in\mathbb{N}}$ *where* $P(C_{i,n}) \sim U[a_i, b_i + 1/n]$ *for* $i = 1, 2, \ldots, |C|$.

Proof. For the random variables $P(C_{1,n})$, $P(C_{2,n})$, …, $P(C_{|C|,n})$, we denote the resulting random variable generated by f as:

$$M_n := f(P(C_{1,n}), P(C_{2,n}), ..., P(C_{|C|,n}))$$

We define analogously the satisfaction degree calculated by the framework as:

$$\overline{M_n^m} = \frac{1}{m}\sum_{k=1}^{m} f(P^k(C_{1,n}), P^k(C_{2,n}), ..., P^k(C_{|C|,n}))$$

By the law of large numbers, it holds:

$$\overline{M_n^m} \overset{m\to\infty}{\longrightarrow} \mathbb{E}[M_n]$$

Given the weak convergence of $P(C_{i,n}) \overset{n\to\infty}{\longrightarrow} P(C_i)$ (refer to Lemma 3.5) and the fact that the satisfaction degree estimator is continuous (refer to Lemma 3.4), it holds:

$$M_n = f(P(C_{1,n}), P(C_{2,n}), \ldots, P(C_{|C|,n})) \overset{n\to\infty}{\longrightarrow} f(P(C_1), P(C_2), \ldots, P(C_{|C|})) = M$$

For the sequence of expected values $(\mathbb{E}[M_n])_{n\in\mathbb{N}}$, it consequently holds:

$$|\mathbb{E}[M_n] - \mathbb{E}[M]| = |\mathbb{E}[M_n - M]| \overset{n\to\infty}{\longrightarrow} 0$$

□

3.6.4. Monotonicity

We study the framework's result under the random variables $P(C_i) \sim U[a_i, b_i], i = 1, 2, \ldots, o, \ldots |C|$, when $P(C_o)$ is exchanged by a random variable $P(C_o)' \sim U[a'_o, b'_o]$ with $a'_o \geq a_o$ and $b'_o \geq b_o$. We say that monotonicity is given if the framework's result is larger under $P(C_i) \sim U[a_i, b_i], i = 1, \ldots, |C|$ than under the same set of random variables where $P(C_o)$ is substituted by a random variable $P(C_o)'$. Formally, this is expressed as follows:

$$\begin{aligned}\mathbb{E}(M') &= \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o)', \ldots, P(C_{|C|}))\ ) \\ &\leq \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o), \ldots, P(C_{|C|}))\ ) = \mathbb{E}(M)\end{aligned}$$

Before proving the main theorem, we define and prove three lemmata.

Lemma 3.7. *The probability that an adversary causes impact i on requirement r in scheme A is non-decreasing with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. Let the set of instantiated capabilities C^A of A be indexed. Let H_j denote the indices of instantiated capabilities appearing in $\alpha_j^{A,r,i}$. Suppose w.l.og. that raising the probability $P(C_o)$ affects the instantiated capability c_h^A. Let

$$I = \{j \mid c_h^A \in \alpha_j^{A,r,i}\}, \quad J = \{j \mid c_h^A \notin \alpha_j^{A,r,i}\}$$

be the indices of minimal cut sets that contain c_h^A (I) and do not contain c_h^A (J). For $j \in I$, let

$$\overline{\alpha_j^{A,r,i}} = \bigwedge_{k\in H_j\setminus h} c_k^A.$$

denote the minimal cut set $\alpha_j^{A,r,i}$ without capability c_h^A. Then the following holds, as similarly shown in [Mat16]:

$$\begin{aligned}\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i} &= \bigvee_{j\in I} \alpha_j^{A,r,i} \vee \bigvee_{j\in J} \alpha_j^{A,r,i} \\ &= \left(c_h^A \wedge \bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \right) \vee \bigvee_{j\in J} \alpha_j^{A,r,i} \\ &= \left(c_h^A \wedge \left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i} \right) \right) \sqcup \bigvee_{j\in J} \alpha_j^{A,r,i},\end{aligned}$$

where $X \sqcup Y$ denotes the disjoint logical disjunction of X and Y. The capability c_h^A does neither appear in $\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}}$ nor in $\bigvee_{j\in J} \alpha_j^{A,r,i}$. Hence, the probability $P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i})$ can be calculated as:

$$P\left(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}\right) = P(c_h^A) \cdot P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) + P\left(\bigvee_{j\in J} \alpha_j^{A,r,i}\right)$$

Because neither $P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right)$ nor $P\left(\bigvee_{j\in J} \alpha_j^{A,r,i}\right)$ are affected by $P(c_h^A)$, it can be concluded that $P\left(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}\right)$ is non-decreasing. □

Lemma 3.8. *The satisfaction degree estimator for requirement r in scheme A is non-increasing with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. According to the security evaluation framework, the risk of requirement r in scheme A and impact level i can be calculated by:

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^i) \cdot \frac{i}{n}$$

The probability $P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^i)$ is continuous with regard to the probability $P(C_o)$ (refer to Lemma 3.3), non-decreasing in probability $P(C_o)$ (refer to Lemma 3.7) and is non-negative. The function $\frac{i}{n}$ does not depend on the probability of $P(C_o)$.

By the following simple argument, it can be shown that the product of two continuous, non-negative and non-decreasing functions f and g, is again a continuous, non-negative, and non-decreasing function h. Because of the monotonicity of f and g, for any $x_1 \leq x_2$,

it holds that $f(x_1) \leq f(x_2)$ and $g(x_1) \leq g(x_2)$. Furthermore, we know that $f(x) \geq 0$ and $g(x) \geq 0$ for all x. Hence, it holds:

$$h(x_1) = f(x_1) \cdot g(x_1) \leq f(x_2) \cdot g(x_1) \leq f(x_2) \cdot g(x_2) = h(x_2)$$

As a result of this inequality, the risk of requirement r in Scheme A is non-decreasing with regard to the probability $P(C_o)$.

Subsequently, from the n computed risk values, the maximum risk is determined. It holds that the maximum of two non-decreasing functions is again non-decreasing [CL12]. By subtracting the maximum risk value from 1, the monotonicity is inverted. Hence, the satisfaction degree estimator is non-increasing.

□

Lemma 3.9. *Let two random variables $X \sim U[a, b]$ and $Y \sim U[c, d]$ with $c \geq a$ and $d \geq b$ be given. For any non-decreasing function f, it holds:*

$$\mathbb{E}[f(X)] \leq \mathbb{E}[f(Y)]$$

Proof. Let Z be a random variable defined as follows:

$$Z = c + (d - c) \cdot \frac{X - a}{b - a}$$

It can be seen that $Z \sim U(c, d)$, and hence particularly Z is equally distributed to Y. Consider the following function:

$$g(x) = c + (d - c) \cdot \frac{x - a}{b - a} - x$$

It holds that $g(a) = c - a \geq 0$ and $g(b) = d - b \geq 0$. Because g is linear, it holds that $g(x) \geq 0$ for all $x \in [a, b]$. Hence, we can conclude that $X \leq Z$. Given the fact that f is non-decreasing, it holds $f(X) \leq f(Z)$ almost surely. From this, we are able to conclude that

$$\mathbb{E}[f(X)] \leq \mathbb{E}[f(Z)] = \mathbb{E}[f(Y)].$$

□

Theorem 3.10. *Let $P(C_i) \sim U[a_i, b_i], i = 1, 2, \ldots, |C|$ denote uniformly distributed random variables for the probabilities of adversarial capabilities C_i. The satisfaction degree of A with regard to requirement r is non-increasing with when random variable $P(C_o)$ is exchanged by $P(C_o)' \sim U[a'_o, b'_o]$, with $a'_o \geq a_o$ and $b'_o \geq b_o$.*

Proof. For $P(C_1), \ldots, P(C_o)', \ldots, P(C_{|C|})$, we denote the resulting random variable generated by f by M', and the respective expected value by $\mathbb{E}[M']$.

By Lemma 3.9 and the fact that the satisfaction degree estimator is non-increasing (refer to Lemma 3.8), we are able to conclude that

$$\begin{aligned}\mathbb{E}(M') &= \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o)', \ldots, P(C_{|C|}))\ ) \\ &\leq \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o), \ldots, P(C_{|C|}))\ ) = \mathbb{E}(M).\end{aligned}$$

□

In fact, the proven monotonicity can be strengthened to strict monotonicity. To prove strict monotonicity of the construction with regard to scheme A and requirement r, the following assumptions are made:

- For each capability $C_o \in C$ with probability distribution $\mathbb{P}_{C_o}$, it holds that $\mathbb{P}_{C_o} \sim U[a, b]$ with $b > 0$.
- For each impact level i and each capability $C_o \in C$, there is at least one instantiation $c_{j,k}^{A,r,i}$ of C_o.
- We say that the probability distribution $P_{C_o} \sim U[a, b]$ increases to $P'_{C_o} \sim U[c, d]$, if $c > a$ and $d > b$.

Informally, the first assumption ensures that strict monotonicity cannot be violated by a capability that is needed for all attack strategies, of which the probability is constantly zero. The second assumption ensures that the maximum risk value is influenced by all capabilities. This assumption is needed to ensure that it cannot happen that the probability distribution of a capability is increased that does not influence the satisfaction degree.

Lemma 3.11. *The probability that an adversary causes impact i on requirement r in scheme A is increasing with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. To prove this lemma, we follow the proof of lemma 3.7 up to the following formula:

$$P\left(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}\right) = P(c_h^A) \cdot P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) + P\left(\bigvee_{j\in J} \alpha_j^{A,r,i}\right)$$

To prove the strict monotonicity of the right-hand term with regard to $P(c_h^A)$, we have to prove that $P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) > 0$. Now suppose this inequality would not hold. This could happen in two cases:

First, this could happen if $\bigvee_{j\in J} \alpha_j^{A,r,i}$ would completely cover $\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}}$. If this would hold, then none of the cut sets in $\left(c_h^A \wedge \bigvee_{j\in I} \overline{\alpha_j^{A,r,i}}\right)$ would be minimal, because they would be completely covered by the cut sets in $\bigvee_{j\in J} \alpha_j^{A,r,i}$.

Second, if the probability of one capability $c_x^{A,r,i}$ in $\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}$ equals 0, it could happen that the term $P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) = 0$. This contradicts, however, the first assumption made for the monotonicity.

Given the fact that $P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) > 0$, we can consequently conclude that $P(c_h^A) \cdot P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) + P\left(\bigvee_{j\in J} \alpha_j^{A,r,i}\right)$ strictly increases, and so also

$$P\left(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^{A,r,i}\right) = P(c_h^A) \cdot P\left(\bigvee_{j\in I} \overline{\alpha_j^{A,r,i}} \setminus \bigvee_{j\in J} \alpha_j^{A,r,i}\right) + P\left(\bigvee_{j\in J} \alpha_j^{A,r,i}\right)$$

strictly increases with regard to the probabilities $P(ec)$ for $c \in C$.

□

Lemma 3.12. *The satisfaction degree estimator for requirement r in scheme A is decreasing with regard to a sample probability $P(C_o)$ for any $C_o \in C$.*

Proof. According to the security evaluation framework, the risk of requirement r in scheme A and impact level i can be calculated by:

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^i) \cdot \frac{i}{n}$$

The probability $P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^i)$ is continuous in the probability $P(C_o)$ (refer to Lemma 3.3), increasing in probability $P(C_o)$ (refer to Lemma 3.11) and is non-negative. The function $\frac{i}{n}$ does not depend on the probability of $P(C_o)$.

By the following simple argument, it can be shown that the product of a continuous, non-negative and increasing function f and continuous, non-negative and non-decreasing function g, is again a continuous, non-negative, and increasing function h.

For any $x_1 \leq x_2$, because of the monotonicity of f and g, it holds that $f(x_1) \leq f(x_2)$ and $g(x_1) \leq g(x_2)$. Furthermore, we know that $f(x) \geq 0$ and $g(x) \geq 0$ for all x. Hence, it holds:

$$h(x_1) = f(x_1) \cdot g(x_1) < f(x_2) \cdot g(x_1) \leq f(x_2) \cdot g(x_2) = h(x_2)$$

Because of this inequality, it can be concluded that the risk value

$$P(\bigvee_{j=1}^{\xi^{A,r,i}} \alpha_j^i) \cdot \frac{i}{n}$$

is increasing with regard to the probability $P(C_o)$.

Because of the second assumption made for the strict monotonicity, the risk of each impact level is increasing, and so is the maximum risk. By subtracting the maximum risk

value from 1, the monotonicity is inverted. Hence, the satisfaction degree estimator is decreasing.

□

Lemma 3.13. *Let two random variables $X \sim U[a,b]$ and $Y \sim U[c,d]$ with $c > a$ and $d > b$ be given. For any non-decreasing function f, it holds:*

$$\mathbb{E}[f(X)] < \mathbb{E}[f(Y)]$$

Proof. Let Z be a random variable defined as follows:

$$Z = c + (d-c) \cdot \frac{X-a}{b-a}$$

It can be seen that $Z \sim U(c,d)$, and hence particularly Z is equally distributed to Y. Consider the following function:

$$g(x) = c + (d-c) \cdot \frac{x-a}{b-a} - x$$

It holds that $g(a) = c - a > 0$ and $g(b) = d - b > 0$. Because g is linear, it holds that $g(x) > 0$ for all $x \in [a,b]$. Hence, we can conclude that $X < Z$. Given the fact that f is increasing, it holds $f(X) < f(Z)$ almost surely. From this, we are able to conclude that

$$\mathbb{E}[f(X)] < \mathbb{E}[f(Z)] = \mathbb{E}[f(Y)].$$

□

Theorem 3.14. *Let $P(C_i) \sim U[a_i, b_i], i = 1, 2, \ldots, |C|$ denote uniformly distributed random variables for the probabilities of adversarial capabilities C_i. The satisfaction degree of A with regard to requirement r is decreasing when the random variable $P(C_o)$ is replaced by $P(C_o)' \sim U[a'_o, b'_o]$, with $a'_o \geq a_o$ and $b'_o \geq b_o$.*

Proof. For $P(C_1), \ldots, P(C_o)', \ldots, P(C_{|C|})$, we denote the resulting random variable generated by f by M', and the respective expected value by $\mathbb{E}[M']$.

By Lemma 3.13 and the fact that the satisfaction degree estimator is decreasing (refer to Lemma 3.12), we are able to conclude that

$$\begin{aligned}\mathbb{E}(M') &= \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o)', \ldots, P(C_{|C|}))\ ) \\ &< \mathbb{E}(\ f(P(C_1), P(C_2), \ldots, P(C_o), \ldots, P(C_{|C|}))\ ) = \mathbb{E}(M).\end{aligned}$$

□

3.7. Summary

Investigating legal provisions for Internet voting systems revealed the lack of a security evaluation framework for Internet voting schemes that quantitatively measures the enforcement of security requirements in specific election settings. To fill this gap, we constructed a security evaluation framework for Internet voting schemes that incorporates the election setting into the evaluation process.

The framework captures a set of legally-founded security requirements for Internet voting schemes and a set of adversarial capabilities. On the one side, the security requirements and the adversarial capabilities serve system analysts to analyze Internet voting schemes qualitatively with regard to their security in an election-independent manner. Furthermore, qualitative security models serve as quality criterion for election-independent improvements of Internet voting schemes. On the other side, the adversarial capabilities allow election officials to specify their election setting in terms of expected adversaries. On the foundation of qualitative security models of an Internet voting scheme and an election setting specification, the framework determines to what extent the scheme enforces the legally-founded security requirements in the specific election setting.

To substantiate the reasonableness and consequently the value of the constructed security evaluation framework, we showed that the framework satisfies a variety of properties borrowed from the context of mathematical measures. In the absence of adversaries, which translates to a probability of 0 assigned to all adversarial capabilities, the framework returns the maximum satisfaction degree for all legally-founded security requirements. It was shown that the satisfaction degree of an Internet voting scheme resistant against specific adversarial capabilities does not change if the adversary changes with regard to those capabilities. Furthermore, the framework proves to be continuous and monotone with regard to adversaries, *i.e.* small increases (respectively decreases) in the adversarial capabilities result in small decreases (respectively increases) of the calculated satisfaction degrees.

Throughout the second part of this thesis, the constructed security evaluation framework will be used to evaluate two established Internet voting schemes, and to propose and evaluate improvements of both schemes.

Part II.

Security Evaluation and Improvement of Internet Voting Schemes

Chapter 4

Foundations for the Evaluation and Improvement of Internet Voting Schemes

Part II of this thesis evaluates the security of two established Internet voting schemes and proposes improvements for both schemes. Both schemes and their respective improvements build upon cryptographic approaches to enforce legally-founded security requirements. Throughout this chapter, we first provide the cryptographic foundations for the remainder of this work. Subsequently, we introduce probabilistic adversaries as basis of the election settings considered in the later evaluations.

The content of this chapter has been published partially as survey in the book *Design, Development, and Use of Secure Electronic Voting Systems* [12] and partially in the journal *Datenschutz und Datensicherheit* [23].

4.1. Cryptographic Primitives and Protocols

We provide the reader with cryptographic primitives and protocols underlying the following Internet voting schemes and their improvements.

Secret Sharing

Secret sharing allows splitting a secret apart such that individual shares do not allow conclusions about the secret but a set of shares allows one to reconstruct the secret.

Specification. A secret sharing scheme is a tuple of algorithms (S, R), where S is the sharing algorithm and R the reconstruction algorithm.

A simple secret sharing scheme can be implemented by the XOR ($\oplus$) operator. Assume a dealer wants to share secret s among n participants. Then the dealer randomly draws $s_1, \ldots, s_{n-1}$ and computes s_n, such that the following equation holds:

$$s = s_1 \oplus \ldots \oplus s_{n-1} \oplus s_n$$

The dealer provides shareholder i with s_i. If all shareholders release their shares, they can reconstruct s according to the above definition. One drawback of this technique is that all shares are needed to reconstruct the shared secret. Hence, the loss of a single share would prevent the secret from being computed.

Shamir / Feldman Secret Sharing. In contrast to the simplest form of secret sharing, a (t, n) *threshold secret sharing* allows reconstructing the secret having $t < n$ shares. Shamir [Sha79] presents a protocol in which the dealer randomly draws values $r_1, \ldots, r_{t-1}$ and generates a polynom of degree t of the following form

$$f(x) = s + r_1 x + r_2 x^2 + \ldots + r_{t-1} x^{t-1}$$

The dealer computes key shares $f(1), \ldots, f(n)$ and provides each participant i with her share $(i, f(i))$ for $i \in \{1, \ldots, n\}$. According to the fundamental theorem of algebra, for an arbitrary t-set of shares $(i, f(i))$, the polynom $f(x)$ can be reconstructed by the Lagrange interpolation:

$$f(x) = \sum_{i=0}^{t-1} f(i) \cdot \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

The secret s is given by the equation $s = f(0)$.

Shamir's scheme relies on a trusted *dealer* that has to split the secret properly; otherwise corrupt shares cannot be identified and composing distinct sets of t shares would result in distinct reconstructed values. In verifiable secret sharing schemes, the dealer has to provide proofs that the issued secret shares allow to reconstruct the secret afterwards. One technique to extend Shamir's scheme has been proposed by Feldman [Fel87]. Assume two large primes q, p are given such that $q|(p-1)$ and a generator g of order q. The dealer, after generating polynom $f(x)$, commits on this polynom by publishing

$$g^s \bmod p,\ g^{r_1} \bmod p,\ \ldots,\ g^{r_{t-1}} \bmod p.$$

Whenever the dealer issues a share to a shareholder i, this shareholder can verify that her share was created in the correct way by checking the

$$g^{f(i)} = g^s \cdot g^{r_1 \cdot i} \cdot g^{r_2 \cdot i^2} \cdot \ldots \cdot g^{r_{t-1} \cdot i^{t-1}} \bmod p.$$

To reconstruct the secret, each shareholder forwards the proof of the dealer such that only correct generated shares are used to reconstruct the secret.

Encryption Schemes

The motivation behind encryption schemes is to encode confidential messages in a way that the code can be transmitted over insecure channels to the intended reader of the message such that this person afterwards can decode the received code to obtain the confidential message.

Specification. Formally, an encryption scheme is a triple of algorithms (G, E, D), where G is a key generation algorithm, E is the encryption algorithm, and D the corresponding decryption algorithm. Encryption schemes can be *asymmetric* and *symmetric*: In the symmetric case, encryption key e and decryption key d are equal and therefore not known to the public, while for asymmetric encryption schemes $e \neq d$ and e is known to the public. Asymmetric encryption schemes can be further classified into *deterministic* and *probabilistic* asymmetric encryption schemes: deterministic schemes map identical messages to identical ciphertexts, as opposed to probabilistic encryption schemes that integrate randomness into the encryption procedure such that two encryptions of identical messages lead to distinct ciphertexts.

There exist a large number of encryption schemes, among which the most important symmetric schemes are *DES* (Data Encryption Standard) and *AES* (Advanced Encryption Standard). The first asymmetric and one of the most influential deterministic asymmetric encryption schemes is *RSA* [RSA78], and well-established probabilistic encryption asymmetric schemes are *ElGamal* [Gam85] and *Paillier* [Pai99].

ElGamal Encryption Scheme. The ElGamal encryption scheme [Gam85] turns out to be of great value for Internet voting schemes due to its important homomorphic properties. Homomorphic cryptosystems allow specific functional operations on plaintexts that result in different functional operations on the corresponding ciphertext. Given two algebraic groups $(P, \oplus)$ and $(C, \otimes)$, then ϕ is a homomorphic mapping between groups $(P, \oplus)$ and $(C, \otimes)$ if for all $p_1, p_2 \in P$, it follows that

$$\phi(p_1 \oplus p_2) = \phi(p_1) \otimes \phi(p_2).$$

As outlined in the following, the homomorphic character of the ElGamal cryptosystems allow to implement a number of operations, such as the re-encryption of ciphertexts.

Key Generation. The key generation algorithm outputs a large prime p and a generator g for the multiplicative group Z_p^*. Furthermore, the algorithm outputs a random number $x \leftarrow \{2, \ldots, p-2\}$ as secret key and $(g, p, y = g^x \pmod p)$ as public key.

Joint Feldman Distributed Key Generation. We present an adaptation by Gennaro *et al.* [GJKR07] of the distributed key generation scheme introduced by Feldman [Fel87]. Goal of this key generation algorithm is to establish a joint public key such that the corresponding secret key is not known to anybody.

1. Participant i generates a polynomial of degree t over Z_q,

$$p_i(x) = a_{i,0} + a_{i,1}x + \ldots + a_{i,t}x^t,$$

where $a_{i,0}$ denotes the shared secret. For each participant j, participant i then computes $x_{i,j} = p_i(j)$ and provides j with that value. Furthermore, i commits on the generated polynomial p_i by publishing the values $X_{i,k} = g^{a_{i,k}}$ for all $0 \leq k \leq t$.

2. Each participant j verifies the shares obtained from all other participants by checking whether the following equation holds:

$$g^{x_{i,j}} = \prod_{k=0}^{t} X_{i,k}^{j_k} \bmod p$$

3. The public value is computed by $y = g^a \cdot \prod_{i=1}^{n} X_{i,0} \bmod p$, while the secret value can be computed as $x = a + \sum_{i=1}^{n} x_{i,0} \bmod p$.

Encryption. Given a public key (g, p, y), a message $m \leftarrow \{0, \ldots, p-1\}$ is encrypted with randomness $r \leftarrow \{2, \ldots, p-2\}$ in the following way:

$$(c_1, c_2) = (g^r, m \cdot y^r) \bmod p$$

Decryption. Given a ciphertext (c_1, c_2) encrypted under public key (g, p, y), message m is reconstructed as follows:

$$m = c_2 \cdot c_1^{-x}$$

Homomorphic Property. The ElGamal encryption scheme provides an important property for Internet voting schemes, namely it is homomorphic. Given two ElGamal ciphertexts $c_i = (g^r, m_1 \cdot y^r)$ and $c_j = (g^s, m_2 \cdot y^s)$ for messages m_1, m_2, it holds that $c_i \cdot c_j$ is a valid ciphertext of message $m_1 \cdot m_2$ as shown below.

$$c = c_i \cdot c_j = (g^r, m_1 \cdot y^r) \cdot (g^s, m_2 \cdot y^s) = \left(g^{r+s}, m_1 \cdot m_2 \cdot y^{r+s}\right) \bmod p$$

For Internet voting, it might be more useful to add messages rather than multiplying them. Therefore, the ElGamal encryption scheme has been extended towards additive homomorphism. The resulting scheme is called Exponential ElGamal [CGS97] and ciphertexts consequently have the following form:

$$(c_1, c_2) = (g^r, g^m \cdot y^r) \bmod p$$

It can easily be seen that the multiplication of individual ciphertexts results in the addition of the underlying plaintexts.

$$c = c_i \cdot c_j = (g^r, g^{m_1} \cdot y^r) \cdot (g^s, g^{m_2} \cdot y^s) = \left(g^{r+s}, g^{m_1+m_2} \cdot y^{r+s}\right) \bmod p$$

It should be noted that decryption of this ciphertext does not immediately result in m, but rather in g^m. Finally, the discrete logarithm of $g^{m_1+m_2}$ must be computed, which is only feasible for small exponents.

Re-encryption. Given a ciphertext $(c_1,\ c_2) =\ (g^r,\ m \cdot\ y^r) \bmod p$ encrypted under public key $(p,\ g,\ y)$, this ciphertext can be re-encrypted using randomness $s \leftarrow\ \{2,\ \ldots,\ p-2\}$ in the following way:

$$\left(c_1',\ c_2'\right) =\ (g^r \cdot\ g^s,\ m \cdot\ y^r \cdot\ y^s) \bmod p$$

The concept of re-encryption is extended to a set of ciphertexts encrypted under the same public key in a straight-forward manner.

Distributed Decryption. So far, the concept of distributed key generation has been abstract. The concept proves, however, to be of great importance to distributed decryption. In distributed decryption, a ciphertext is partially decrypted by participants such that the partial decryption can be used to reconstruct the plaintext based on the Lagrange interpolation. Let an ElGamal ciphertext $c = (c_1, c_2)$ be given. Throughout the decryption phase, participant i applies her secret key share x_i and computes her partial decryption

$$c_1\,(i) = c_1^{x_i}$$

and publishes a proof showing that

$$c_1\,(i) = x_i =\ y_i.$$

If the participant's proof does not convince the majority of participants, they decide to reconstruct her secret key share in a distributed way relying on the Lagrange interpolation of the committed shares of the secret key shares of participant i. The honest participants are capable of reconstructing x_i and hence $c_1\,(i) = c_1^{x_i}$.

Once, all participants' partial decryptions $c_1\,(i)$ are available, the plaintext is reconstructed as:

$$m = \frac{c_2}{\prod_{i=1}^{n} c_1\,(i)}$$

Digital Signatures

The goal of signature schemes is to ensure the integrity and authenticity of messages with respect to the sender as well as non-repudiation.

Specification. A signature scheme is a triple of algorithms (G, S, V), where G is a key generation algorithm, S is the signing algorithm, and V the verification algorithm.

RSA Signature. The key generation, signing, and verification processes of the RSA signature scheme are described.

Key Generation. Given two large primes $p,\ q$, two values $n = p \cdot q$ and $\varphi(n) = (p-1) \cdot (q-1)$ are computed. A value e with $1 < e < \varphi(n)$ co-prime to $\varphi(n)$ is randomly chosen and d is determined such that

$$e \cdot d \equiv 1 \bmod \varphi(n).$$

The verification key is (e, n), the signing key is d.

Signing. Given the signing key d, a message $m < n$ is signed according to the following equation:

$$s = m^d \bmod n$$

Verification. Given a verification key (e, n), signature s on message m is valid if the following equation holds:

$$s^e = m \bmod n$$

Zero-Knowledge Proof Systems

Zero-knowledge (ZK) proof systems are the cryptographic tool to prove the validity of statements without revealing anything beyond the validity of this statement.

Specification. A ZK proof system is given by a tuple of algorithms (P, V), where P is the prover of statements and V is the verifier of these statements. A ZK proof system for given language L satisfies three properties: 1) each valid statement can be proven (completeness), 2) no invalid statements can be proven (soundness), a malicious verifier does not learn anything beyond the validity of the statement (zero-knowledge). In the context of Internet voting, there exist numerous specific ZK proofs, *e.g.* designated-verifier proofs, proof of equality of discrete logarithms, 1-out-of-L encryption proofs, disjunctive proof of equality between discrete logarithms. We refer the interested reader to Smith [Smi05] for detailed information.

Proof of Knowledge of Discrete Logarithm. Schnorr [Sch90] invented a protocol to prove the knowledge of discrete logarithm. Given basis $g \leftarrow Z_p$, value $y \leftarrow Z_p$, the prover wants prove that she knows l such that $y = g^l$ where g and y are publicly known. The protocol is summarized as follows:

1. The prover randomly draws $r \leftarrow Z_p$ and outputs $a = g^r$
2. The verifier randomly draws $c \leftarrow Z_p$ and outputs c

3. The prover computes $z = r + l \cdot c$ and outputs z

4. The verifier checks if $g^z = a \cdot y^c$

Homomorphic Tallying

One typical application of homomorphic encryption schemes are Internet voting schemes that implement homomorphic tallying. In that approach, rather than decrypting individual votes in the tallying phase, first the encrypted sum of all encrypted votes is computed and thereafter decrypted. Due to the homomorphism of the encryption scheme, the sum of encrypted votes equals the encryption of the sum of votes, *i.e.* the encryption of the election result. Thereby, neither the public, nor any service provider learn complete votes as cast by voters.

In the simplest case of referendum (Yes/No election), homomorphic encryption schemes can be implemented in a straightforward manner. First voter i makes her selection $v_i \in \{0, 1\}$ and encrypts her selection with the public election key pk distributively shared between independent service providers. The voter thereafter binds her authentication data to her encrypted vote, *e.g.* by posting her name together with $\{v_i\}_{pk}^{r_i}$ on the bulletin board. The voter furthermore provides a ZK proof that her vote is a valid vote in order to prevent malicious voters from over-voting (*i.e.*, a proof showing that $v_i \in \{0, 1\}$). The voter can convince herself that her vote was stored in an unaltered way on the bulletin board by checking if her name appears next to her encrypted vote and the corresponding proof.

In the tallying phase, the public can calculate the encrypted result by multiplying the encrypted individual votes.

$$R = \{v_1\}_{pk}^{r_1} \cdot \ldots \cdot \{v_n\}_{pk}^{r_n} .$$

The result can be computed by decrypting the product R with the corresponding secret key; hence

$$r = D\,(sk, R) .$$

Finally, the service providers prove that they properly decrypted, *i.e.* that they decrypted the product of encrypted votes with the proper secret shares by generating a ZK proof of correct decryption based on a ZK proof of equality of discrete logarithms.

Iterative Cut-and-Choose Verification

If an homomorphic encryption scheme is in place to encrypt votes, malicious voting devices might encrypt votes differently than intended by the voters. Assume a user intends to encrypt message m with a public encryption key pk using the ElGamal encryption scheme

in an arbitrary Internet voting scheme. Then, in accordance to the encryption algorithm, the system draws randomly $r \leftarrow \{2, \ldots, p-2\}$ and computes

$$(c_1, c_2) \;=\; (g^r, m \cdot\; y^r) \mod p.$$

The question arises how the user can be sure that the system encrypted the right value, anyway the output will be indistinguishable by definition for all input values. To counter this threat, Benaloh [Ben06] proposed a concept to prove the integrity of probabilistic vote encryptions in a ZK proof manner: After encrypting m, the system commits on the encryption process by providing the user with a cryptographic hash $H(c_1, c_2)$. The user thereafter (unpredictably) decides if she audits or accepts the encryption process of the device. If she decides to audit the process, the device returns the randomness r. The user can verify the correct encryption by computing $(c_1', c_2') \;=\; (g^r, m \cdot\; y^r) \mod p$ locally or with the help of an external institution and checks whether $H(c_1, c_2) = H(c_1', c_2')$. After the verification process, the voter cannot use the audited ciphertext because, depending on the scheme, the obtained randomization factor could serve as proof about her vote. The voter might consequently become target of attacks or might sell her vote to vote buyers. The voter can repeat the cut-and-choose verification process an arbitrary number of times. Once, the voter is convinced about the fact that the voting device correctly encrypts the actual vote, the voter does not audit the encryption process and casts the encrypted vote.

Code Voting

The concept of code voting goes back to Chaum's SureVote [Cha01]. The goal of code voting is to enforce vote secrecy against compromised voting devices. To deploy code voting, in the pre-voting phase, unique code sheets for all eligible voters are generated: a code sheet contains the code sheet ID and a two-column table, where each candidate has a voting code assigned. A typical code sheet is shown in Figure 4.1. After their generation,

Code Sheet ID: 34255	
Candidate	Voting Code
Alice	51948
Bob	23766
Eve	41948

Figure 4.1: Code sheet for the application of code voting.

the code sheets are assigned and issued to voters and furthermore issued to the service provider in charge of collecting votes. The voter must not receive her code sheet over her voting device, to prevent the voting devices from learning valid voting codes. In the voting phase, the voter casts her vote by sending the code sheet ID and the voting code next to the preferred candidate to the service provider in charge of collecting votes. In case a voter, who possesses the code sheet shown in Figure 4.1, intends to cast a vote for Alice,

she would submit the ballot ID, namely 34255, and the voting code next to Alice, namely 51948. The service provider re-interprets the code, identifies the selected candidate and stores a vote for that candidate. Because a compromised voting device does not know which candidate is represented by the voting code, the voting device cannot break vote secrecy.

Return Codes

Return codes represent the vote integrity-enhancing counterpart of code voting. Consider again the code sheet shown in Figure 4.1. In the case of return codes, the voting codes play the role of return codes, resulting in a sheet as shown in Figure 4.2. In order to cast her vote, depending on the concrete scheme, the voter expresses her preference either directly on her voting device via the user interface or via a voting code in the case of code voting. After the voter has expressed her preference, the vote is transmitted to the central system components which interpret the supposed voter intention and determine the corresponding return code. This code is subsequently returned to the voter. Because of the fact that the voting device does not know the voter's return codes, the device might only show the received return code, *i.e.* the return code associated to the cast intention. In case the device tampered with the vote, the device is not able to provide the voter with the expected return code.

Code Sheet ID: 34255	
Candidate	Return Code
Alice	71468
Bob	53286
Eve	35468

Figure 4.2: Code sheet for the application of return codes.

Independent Verification Devices

One further approach to prevent compromised voting devices from vote tampering is to incorporate logically independent devices for verifying the correctness of the vote encryption process. Since 2013, this approach is implemented in the Estonian Internet voting scheme [HW14]. We explain the use of independent verification devices on the Estonian example. After the voter has expressed her intention over the user-interface, the voting device encrypts the designated voter intention with a probabilistic encryption algorithm. Analogously to the Benaloh challenge approach, the device temporarily stores the used encryption randomness. The signed ciphertext is transmitted to the central components of the voting system. The system assigns a unique identifier to the received ciphertext and stores the signed ciphertext under the respective identifier. The unique identifier is returned to the voting device which presents the identifier and the randomness in form of a

quick response (QR) code to the voter. The voter has the possibility to read the QR code with an application ran on an independent verification device, *e.g.* a smart phone. The application decodes the QR code and queries the central system to release the signed ciphertext stored under the transmitted identifier and all voting options. Upon reception of the data, the verification application on the independent verification device exploratively encrypts all voting options with the randomness provided by the voting device and compares the resulting ciphertexts with the ciphertext received from the central system. After one voting option results in a match of ciphertexts, the respective voting option is shown to the voter. The voter decides whether the output voting option corresponds to her intention.

4.2. Probabilistic Adversaries

In preparation for the quantitative security evaluation within the following chapters, we construct four probabilistic adversaries. A summary of the adversaries is provided in Table 4.1[10].

In the remainder of this work, we consider only adversaries that have full control over the communication channel between voting devices and service providers and between service providers. Hence, all adversaries have the capability *CCH* with a probability of 1 (U[1, 1]). Furthermore, we do not consider adversaries that are computationally unrestricted, as this mainly concerns the functions layer of Internet voting systems. Hence, adversaries have the capability *CR* with a probability of 0 (U[0, 0]). Both capabilities are therefore not explicitly outlined throughout the evaluation.

The first adversary compromises with relatively high probability (U[0.01, 0.1]) either one voting device or is able to control the voter in any sense (sending objects/data to the voter, receiving objects/data from the voter, or controlling the channel between voter and voter device). With lower probability (U[0.001, 0.002]) the adversary is able to compromise an online service provider and with the lowest probability (U[0.0001, 0.0002]) the adversary succeeds in compromising an offline service provider. Building upon the first adversary, we define three adversaries with dedicated strengths.

The second adversary is a reinforcement of the first adversary with regard to voting device corruption. As opposed to the distribution U[0.01, 0.1], the second adversary compromises voting devices with a probability between 0.1 and 0.2 (U[0.1, 0.2]).

The third adversary is particularly strong with regard to compromising either online service providers (reinforced from U[0.001, 0.002] to U[0.01, 0.02]) or offline service providers (reinforced from U[0.0001, 0.0002] to U[0.001, 0.002]).

Furthermore, we define one adversary that is reinforced (from U[0.01, 0.1] to U[0.1, 0.2]) with regard to influencing voters, either in sending objects or data to the adversary, receiving objects or data from the adversary, or controlling the channel between a voter

[10] $U[a, b]$ refers to the uniform distribution with support a and b.

and her voting device(s).

	VD	ONSP	OFSP	VO	VI	HCH
E_1	U$[0.01, 0.1]$	U$[0.001, 0.002]$	U$[0.0001, 0.0002]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$
E_2	U$[0.1, 0.2]$	U$[0.001, 0.002]$	U$[0.0001, 0.0002]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$
E_3	U$[0.01, 0.1]$	U$[0.01, 0.02]$	U$[0.001, 0.002]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$	U$[0.01, 0.1]$
E_4	U$[0.01, 0.1]$	U$[0.001, 0.002]$	U$[0.0001, 0.0002]$	U$[0.1, 0.2]$	U$[0.1, 0.2]$	U$[0.1, 0.2]$

Table 4.1: On the basis of one adversary model, we define three adversary models with dedicated strengths.

Chapter 5

The Polyas Internet Voting Scheme as Applied for the GI 2011 Elections

The Polyas Internet voting scheme has been invented in 1996 and since then is provided by the German companies Micromata Ltd.[11] and Polyas Ltd.[12] To date, the scheme has been used to conduct numerous Internet-based elections in a variety of contexts [OSV11]. Among those, there are University elections, *e.g.* the elections of the council of the Graduate Academy at the Friedrich-Schiller-Universität Jena, elections in companies, *e.g.* the Found Board elections of the SwissLife, and elections in private associations, *e.g.* the Review Board elections of the German Science Foundation. With approximately 2,2 millions cast votes [NVS+15], the Polyas Internet voting scheme represents one of the most established Internet voting schemes in the German-speaking world.

The first section of this chapter provides an overview about the Polyas scheme as applied for the elections held by the Gesellschaft für Informatik e.V. (GI, *engl.* German Informatics Society), in the year 2011. We determine qualitative security models of the original scheme in the second section of this chapter. Based on the qualitative security models, we propose an improvement of the scheme with regard to vote integrity in the third section. Afterwards, we determine the qualitative security models of the extended scheme. Subsequently, we first compare the qualitative security models of both schemes, before the security of both schemes is quantitatively compared in different election settings. The chapter is concluded with a summary of our findings.

Parts of this chapter have been published in the journal *Datenschutz und Datensicherheit* [23].

5.1. Original Scheme

The components involved in the voting process are outlined in the first part of this section. Afterwards, we describe the protocol underlying the voting scheme as applied for the GI

[11] `https://www.micromata.de/en/home/`
[12] `https://www.polyas.de/`

2011 elections.

5.1.1. Components

The Polyas voting scheme comprises the following components:

Generation Server (GS). Due to the lack of publicly available information regarding its implementation, we describe the *GS* as an abstract component. Given the abstract interpretation, we do not consider the *GS* throughout the security evaluation. The *GS* is in charge of generating voting credentials for all eligible voters and providing these credentials to the printing service, the electoral registry server, and the validation server.

Printing Service (PS). The *PS* is in charge of printing and forwarding the voting credentials to the voters. The *PS* operates in an offline manner.

Electoral Registry Server (ERS). In mutual control with the validation server (the *VS*), the *ERS* processes voters' queries and provides voters with voting tokens upon successful verification of their eligibility. The *ERS* operates in an online manner.

Validation Server (VS). In mutual control with the *ERS*, the *VS* processes voters' queries and provides voters with voting tokens upon successful verification of their eligibility. The *VS* operates in an offline manner.

Ballot Box Server (BBS). The *BBS* provides voters with the digital ballot, collects filled ballots, and stores these ballots throughout the voting phase. The *BBS* operates in an online manner.

Tallying Component (TC). The *TC* is in charge of generating the cryptographic key material for the election and decrypting encrypted votes in the tallying phase. The *TC* operates in an offline manner.

Voting Device (VD). Each voter has one voting device at her disposal over which she fills the digital ballot and casts her vote.

5.1.2. Ballot of the GI 2011 Elections

In the GI 2011 elections, two races were held simultaneously[13], namely the election of the presiding council and the election of the management board.

Within the election of the presiding council, five candidates were available. Each voter had the possibility to vote for at most three candidates. In addition, each voter had the

[13]Refer to the GI 2011 elections website: `https://www.gi.de/index.php?id=4165`

possibility to spoil her ballot, either by over-voting or by explicitly selecting the *spoil ballot* option.

Within the election of the management board, four candidates were available. Each voter had the possibility to give a *yes* or *no* vote for each candidate, or not expressing her preference for a candidate at all. Additionally, each voter had the option to spoil her ballot, either by giving a *yes* and *no* vote for any candidate or by explicitly selecting the *spoil ballot* option.

5.1.3. Protocol Description

We present the protocol conducted between the voter and different components of the Internet voting scheme. Throughout the protocol description, we consider an extension of the protocol proposed by Olembo *et al.* [OSV11], which improves the scheme towards integrity by introducing partial verifiability into the scheme. For the ease of readability and to delineate previous contributions from our own contributions, we will refer to this adapted scheme as the *original* Polyas scheme. We consider components of the scheme to operate in independent manner. To propagate this aspect to the system layer, the components have to be implemented and hosted by independent providers.

Setup Phase. All involved service providers generate SSL/TLS and signature keys and publish the respective public keys. Furthermore, the public keys of service providers that interact with voters are provided to the voters[14]. The *GS* generates voting TANs (*transaction numbers*) for all eligible voters. Each voting TAN is subsequently cryptographically hashed and assigned to one voter ID, *i.e.* the identification number over which the voter is identifiable by the election holding association, implemented by the *ERS*. The voter IDs and the respective hashed voting TANs are subsequently transmitted for further usage to the *ERS*. Additionally, the hashed voting TANs are transmitted for further usage to the *VS*. Together with voting TANs, the names and addresses of voters to which these TANs are assigned, are forwarded to the *PS*. The *PS* prepares voting materials for all eligible voters, packs those materials into sealed envelopes, and sends those envelopes to voters via postal mail. The *TC* generates an asymmetric election key pair (pk, sk) and sends the public election key to the *BBS*. The secret key is stored within the *TC*. Furthermore, the digital ballot is stored within the *BBS*. The protocol steps of the Polyas setup phase are depicted in Figure 5.1.

Voting Phase. Upon reception of the voting materials, the checks checks integrity of the sealed envelopes. To start the voting process, the voter visits the website of the *ERS*. The voter authenticates herself by providing her ID and her voting TAN to the *ERS*. The *ERS* hashes the received voting TAN and matches the hash value against the entry for the

[14]For the ease of readability, we omit the key exchange in the sequence diagram. For the same reason, we omit the fact that channels between voting device(s) and service providers are authentic and confidential.

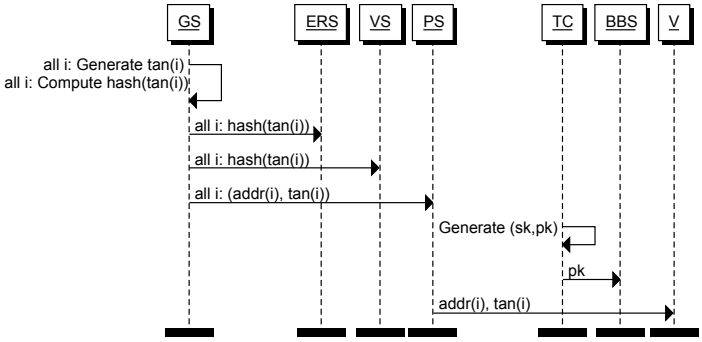

Figure 5.1: Setup phase of the original Polyas Internet voting scheme.

respective voter in the internal database. If this match succeeds, the server forwards the received voting TAN to the *VS*. Analogously to the *ERS*, the *VS* hashes the received voting TAN and compares the hash value against the internal database. If this check succeeds, the *VS* generates a random voting token for the respective voting TAN. The voting token is subsequently forwarded to the *BBS* and the *ERS*. The latter forwards the voting token to the voter and in turn forwards the voter to the *BBS*. The voter presents her voting token. The *BBS* checks the voter's eligibility by verifying whether the presented voting token has previously been issued by the *VS*. If so, the *BBS* issues the ballot to the voter. The voter fills the ballot according to her preferences and returns the filled ballot to the *BBS*. For the sake of re-assurance, the *BBS* returns the filled ballot to the voter, encrypts the ballot with the public election key, and caches the resulting ciphertext. If the voter confirms her selection, the *BBS* deletes the voting token and stores the encrypted ballot for the tallying phase.[15] For the purpose of improving vote secrecy, the *BBS* stores the ballot in a disassembled manner, as proposed by Olembo *et al.* [OKNV12]. The protocol steps of the Polyas voting phase are depicted in Figure 5.2.

Tallying Phase. After the voting phase has been terminated, the encrypted ballots stored within the *BBS* are transmitted to the *TC*. The authorities hosting the *TC* initiate the decryption process for all encrypted votes that were confirmed and cast by eligible voters. The *TC* decrypts these votes. Ultimately, the number of registered voters in the *ERS* and the *VS*, the encrypted votes stored by the *BBS*, and the decryption key of the *TC* are published. Any observer can use the tool developed by Olembo *et al.* to partially verify the election. The tool decrypts the encrypted votes and provides the user with the obtained election result. The user compares the result obtained from the verification tool to the result announced by the *TC*. The protocol steps of the Polyas tallying phase are depicted in Figure 5.3.

[15] The original Polyas definition foresees the incorporation of a hash chain mechanism. However, this mechanism allows the *ERS* to violate vote secrecy. For a specific voter, the *ERS* can link that voter's

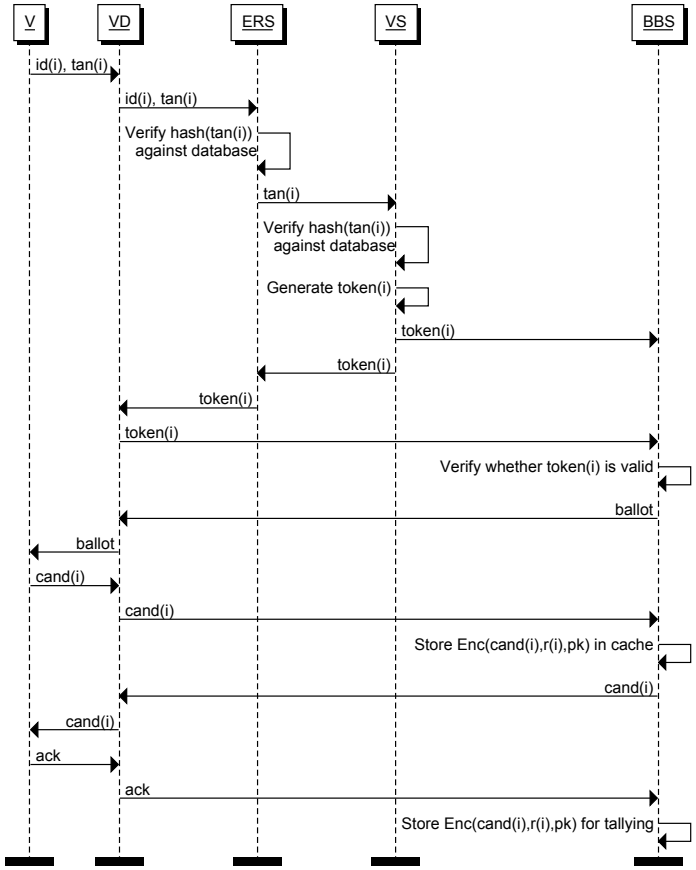

Figure 5.2: Voting phase of the original Polyas Internet voting scheme.

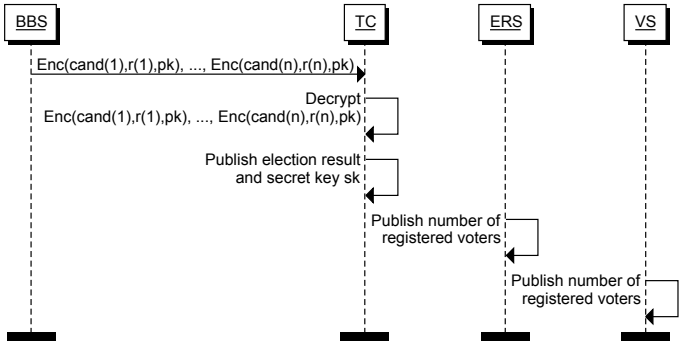

Figure 5.3: Tallying phase of the original Polyas Internet voting scheme.

vote to a set of 30 votes.

5.2. Qualitative Security Models of the Original Scheme

On the foundation of Polyas scheme descriptions [RJ07, MR10, VG09, OSV11] and informal protocol analysis, we take the role of system analysts and determine qualitative security models of the Polyas Internet voting scheme. In the remainder of this section, we outline which capabilities an adversary allow to cause impact on security requirements. An overview of the result is given in Table 5.1. Recall that we make the general assumption that anything that can be verified is verified.

Vote Secrecy. Corrupting or controlling the following components and/or channels allows the adversary to violate vote secrecy.

Voting Device. The voting device knows the voter's identity. Additionally, throughout the voting phase, the voter enters her preference on her voting device. Hence, the voting device is able to relate the voter's identity to her preference. As a result, compromised voting devices can violate vote secrecy of the vote cast over that devices.

Channel between Voter and her Voting Device. The Polyas scheme does not incorporate measures upholding vote secrecy of voters that are observed during the vote casting process. Consequently, an adversary observing the channel between the voter and her voting device is able to violate vote secrecy of one vote.

Printing Service, Validation Server, and Ballot Box Server. Throughout the voting phase, the *VS* learns the relation between a voter's voting TAN and her voting token. On the one side, the *VS*'s knowledge can be combined with the *PS*'s knowledge which assembles the relation between voters' identities and their respective voting TANs to establish the link between the voter identity and the voting token. On the other side, the *VS*'s knowledge can be combined with the *BBS*'s knowledge to incorporate the relation between the voting token and the selected voting option. Consequently, the malicious collaboration of all three components results in the violation of vote secrecy. The malicious collaboration of these components results in a violation of vote secrecy of all votes.

Electoral Registry Server and Ballot Box Server. After the voter has been registered by the *ERS* and obtained a voting token generated by the *VS* throughout the voting phase, the *ERS* knows the relation between the voter's identity and her voting token. The voter uses that token to cast her vote to the *BBS*, thereby proving her eligibility to the *BBS*. Consequently, maliciously combining the *BBS*'s knowledge with knowledge of the *ERS* would lead to the violation of vote secrecy of all votes.

Vote Integrity. Corrupting the following components allows the adversary to violate vote integrity.

Voting Device. The original Polyas scheme does not incorporate mechanisms that prevent or make the violation of vote integrity by compromised voting devices detectable. As a

result, a compromised voting device can alter votes cast over that device undetectably.

Computationally unrestricted and Communication Channel. An adversary controlling the channel between the voting device and the *BBS* and furthermore capable of breaking cryptographic primitives can violate the security guarantees of authenticated and confidential communication channels. Hence, the adversary is able to intercept and alter in an undetectable manner the communication between the voting device and the *BBS*.

Ballot Box Server. The *BBS* is able to alter votes undetectably right after votes have been submitted by voters and before storing them in the ballot box. Thereby, the *BBS* can violate integrity of all votes.

Eligibility. Corrupting or controlling the following components and/or channels allows the adversary to violate eligibility.

Voter Output. In the setup phase, each voter receives her voting credentials, *i.e.* the voting TAN which she can use together with her voter ID to cast her vote. If voters are under adversarial influence and try to forward their right to vote, they can do so by forwarding their voting credentials. Hence, a voter forwarding her voting credentials to the adversary enables the adversary to cast one ineligible vote.

Electoral Registry Server and Validation Server. The *ERS* and the *VS* control each other to some extent. If the *ERS* does not forward a valid voting TAN to the *VS*, the *VS* does not generate a voting token. If on the other side, the *VS* forwards a voting token to an ineligible voter without that voter having presented a valid voting TAN to the *ERS*, the *ERS* detects a discrepancy between the number of registered voter and the number of cast votes which allows the *ERS* to pinpoint the voting token illegitimately generated by the *VS*. However, if both components maliciously cooperate, they are able to cast votes for abstaining voters, thereby violating eligibility of all voters abstaining from the election.

Electoral Registry Server and Printing Service. The *ERS* can circumvent the *VS*'s collaboration to violate eligibility if it knows valid voting TANs for all abstaining voters. The *PS* knows these voting TANs. Compromising the *PS* and the *ERS* allows an adversary consequently to cast votes for abstaining voters, thereby violating eligibility.

Fairness. Corrupting or controlling the following components and/or channels allows the adversary to violate fairness.

Voting Device. Analogously to the vote secrecy case, voting devices learn voters' selection throughout the voting phase. As a consequence, compromised voting devices are able to learn partial election results, thereby violating fairness of the vote cast over that device.

Channel between Voter and her Voting Device. Analogously to the vote secrecy case, a voter observed throughout the voting phase is not able to override her selection. Hence,

an adversary observing the channel between the voter and her voting device is able to violate fairness of one vote.

Ballot Box Server. Throughout the voting phase, the voter casts her vote via the *BBS*'s web-interface to the *BBS*. In spite of the fact that the connection between the voting device and the *BBS* is secured, the *BBS* receives the plaintext vote. Hence, a compromised *BBS* is able to deduce complete intermediate results, thereby violating fairness.

Computationally unrestricted and Communication Channel. If an adversary is capable of controlling the communication channel between the voting device and the *BBS* and is capable of breaking cryptographic primitives, then the adversary can determine the content of encrypted messages, thereby violating fairness.

Data Access Protection. The following components are capable of causing violations of data access protection.

Voting Device. Because of the fact that voting devices are generally used for a number of purposes, voting devices know the voters' identities. A compromised voting device might consequently forward a voter's identity to the adversary.

Electoral Registry Server. In advance to the election, the *ERS* stores the electoral register including voters' IDs and hashed voting TANs. In case of corruption, the *ERS* can provide voter data to the adversary.

Printing Service. The *PS* is in charge of providing voters with voting materials. Hence, the *PS* learns both voter identities and voting TANs. If the *PS* is compromised, it can provide voter data to the adversary.

Discussion. The qualitative security models reveal a variety of shortcomings of the Polyas scheme.

In its first version [RJ07], to enforce vote integrity, the Polyas scheme relies on the assumption that the *TC* correctly decrypts votes throughout the tallying phase. Olembo *et al.* [OSV11] addressed this limitation by incorporating verifiability mechanisms. In spite of the fact that the measure proposed by Olembo *et al.* mitigates the risk of vote integrity violations, the enforcement of vote integrity builds upon the assumption that voting devices are not compromised. In fact, it turns out that four out of five legally-founded security requirements build upon the assumption that voting devices are not compromised. The criticality of this assumption is more and more prevalent both from a technical and legal perspective. According to the quarterly PandaLabs security report [Pan15], more than 32% of computers worldwide are infected by malware. The criticality of infected personal voting devices has also been emphasized within a recent report by WebRoots Democracy [Web16]. Legislators start taking this aspect into account and release corresponding legal regulations. For instance, regulations released by the Swiss Federal Councillor put their focus on the introduction of verifiable voting systems for the purpose of increasing vote

integrity[16]. Primarily this focus refers to the voters' possibility to detect manipulations on their own vote, *i.e.* individual verifiability. Given the fact that the Polyas scheme is currently under Common Criteria certification, the herein proposed extension shall maintain processes, and particularly voter processes, as much as possible. Due to this practical constraint and the general tendency towards improving vote integrity with regard to compromised voting devices, our extension targets at enforcing vote integrity in the presence of compromised voting devices. Hence, we incorporate a verifiability mechanism into the Polyas voting scheme. The mechanism shall allow voters to detect vote manipulations by compromised voting devices.

Requirement	Qualitative Security Models	Impact
Vote Integrity	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee BBS \vee (CR \wedge CCH)$	$1 \leq l \leq n$
Eligibility	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\right) \vee (ERS \wedge VS) \vee (ERS \wedge PS)$	$1 \leq l \leq n$
Fairness	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} HCH_i)\right) \vee BBS \vee (CR \wedge CCH)$	$1 \leq l \leq n$
Vote Secrecy	$(ERS \wedge BBS) \vee (PS \wedge VS \wedge BBS) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} HCH_i)\right) \vee$	$1 \leq l \leq n$
Data Access Protection	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee PS \vee ERS$	$1 \leq l \leq n$

Table 5.1: Qualitative security models of the original Polyas Internet voting scheme.

5.3. Addressing Vote Integrity Vulnerabilities Caused by Compromised Voting Devices

The qualitative security models of the original Polyas scheme unveil that vote integrity is threatened in the presence of compromised voting devices. This section is dedicated to the improvement of the original Polyas voting scheme in that regard. We therefore first review related works on the improvement of the Polyas Internet voting scheme. Subsequently, we review approaches to prevent successful integrity violations caused by compromised voting devices (refer to Section 4.1) and determine the most appropriate approach for the Polyas scheme. Eventually, we show how the approach is embedded into the Polyas voting scheme.

[16]Refer to `https://www.admin.ch/opc/en/classified-compilation/20132343/index.html`

5.3.1. Related Work

In preparation of this work, all works that cite the original Polyas paper [RJ07] have been sight. Three works propose an extension of the original Polyas system [MR10, OSV11, OKNV12]. Menke and Reinhard [MR10] introduce one further component into the original Polyas system, the Committee Tool. The tool fundamentally implements trust distribution on the authority layer above Internet voting schemes. The extension is therefore not relevant to our contribution. The extensions proposed by Olembo *et al.* [OSV11, OKNV12] have been incorporated into our description of the original Polyas scheme.

5.3.2. Feasibility of Cast-as-Intended Verifiability Approaches for Polyas

To extend the Polyas Internet voting scheme towards the enforcement of vote integrity in the presence of compromised voting devices, the feasibility of different approaches to deploy cast-as-intended-verifiability (refer to Section 4.1) is investigated. Thereby, special attention is given to the constraint that extending the Polyas scheme towards cast-as-intended verifiability shall maintain processes and voter experiences as much as possible.

Independent verification devices. The application of independent verification devices requires the voter to possess independent computing devices, *e.g.* a smartphone. This extension would change the voter experience significantly. Furthermore, independent software solutions for these devices have to be developed and provided to the voters. We therefore do not consider the incorporation of independent verification devices into the Polyas scheme.

Iterative cut-and-choose-verification. From a cryptographic perspective, the application of iterative cut-and-choose verification is a promising approach and has found its application in several elections running the Helios voting scheme. The advantage of this approach is that besides vote integrity, the underlying encryption process by its very nature also enforces vote secrecy to a high degree: a voter does neither obtain a receipt about her vote nor do any central components learn the voter's intention. In spite of the fact that vote verification shall be conducted on an independent verification device, one further substantial shortcoming of this approach is the significant involvement of voters in the vote verification process; voters shall conduct vote verification an unpredictable number of times to achieve reasonable integrity assurance. Because both facts fundamentally change the vote casting process, cut-and-choose verification does not comply with the constraint that processes and voter experiences shall largely remain untouched from the extension.

Return codes. We opt for adapting return codes as means improve vote integrity with regard to compromised voting devices. This decision is substantiated by the fact that on the one hand, no special hardware or software is required on the voter-side, the scheme adaption is relatively small, and the use of simple codes for the purpose of security is relatively well-known, *e.g.* from the online banking context and the increasing application

of two-factor authentication. Furthermore, this decision is justified by the fact that voters in the GI setting authenticate themselves with TANs. Consequently, we expect the use of short codes for the sake of vote verification a relatively small adaptation for voters.

5.3.3. Deployment of Cast-as-Intended Verifiability in Polyas

For the integration of return codes into the Polyas scheme, the setup and voting phases are slightly modified.

Setup Phase. Prior to the election, the *ERS* generates for each voter an ordered list of random codes, such that each code can be associated to one voting option. Additionally, the *ERS* generates for each voter one offline authentication code. All random codes are encrypted with the public key of the *BBS* and stored for further processing throughout the voting phase. The *ERS* forwards these codes to the *PS* which associates one list of ordered codes plus offline authentication code to each voter. The codes are printed, while the offline authentication code is printed under a scratch field. In addition to the authentication material (see Section 5.1.3), the codes are issued to the voter. The revised setup phase of the Polyas scheme is depicted in Figure 5.4.

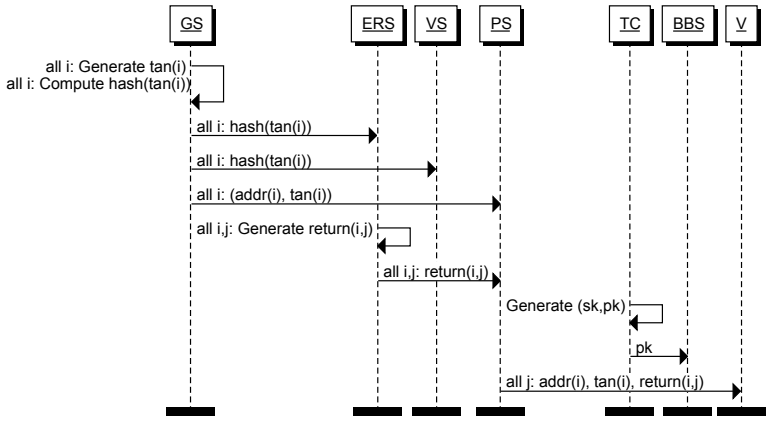

Figure 5.4: Setup phase of the extended Polyas voting scheme.

Voting Phase. After authenticating towards the *ERS* and the *VS*, in addition to a voting token the voter receives the list of encrypted return codes and the offline authentication code generated by the *ERS* within the setup phase. In addition to her voting option, the voter forwards the list of encrypted codes to the *BBS*. After interpreting the supposed voter intention, the *BBS* decrypts the list of encrypted codes and returns the voter the return code(s) assigned to the interpreted voting intentions. The return codes are shown in editable text fields, such that voters can easily override these codes if necessary, *e.g.* if an adversary requests them to prepare a screenshot with a specific return code. To support

the cast-as-intended verification process, the *ERS* preserves the link between voter and assigned voting token, the *BBS* preserves the link between voting token and the cast ballot. Upon reception of the return code(s), the voter matches the code(s) against the return code(s) assigned to her voting option(s) on her code sheet. In the case of the GI 2011 election, voters might distribute at most eight votes within the two ballots. Consequently, the voter would receive at most eight return codes, one for each individual selected option. If the codes match, the voter is convinced that her intention has not been tampered with and arrived in an unaltered manner. The *ERS* and the *BBS* discard the preserved links after a specified time and the *BBS* stores the encrypted ballot for the tallying phase. If the codes do not match, the voter has the possibility to consult a service center of the authority running the *ERS*. Therefore, the voter opens her offline authentication code (which is printed under a scratch field), authenticates towards the service center by means of the authentication code and requests to remove the vote cast in her name. The *ERS* identifies the voting token associated to the voter and sends a queries the *BBS* to remove the vote assigned to the respective voting token. The *ERS* reactivates the voting process to the voter. The revised voting phase of the Polyas scheme is depicted in Figure 5.5.

5.4. Qualitative Security Models of the Extended Scheme

In analogy to Section 5.2, we take the role of system analysts and investigate to which extent the qualitative security models of the original Polyas scheme are affected by the proposed extension. Recall that we make the general assumption that anything which can be verified is verified.

A summary of the qualitative security models of the extended Polyas scheme is presented in Table 5.2.

Vote Secrecy. The introduction of return codes provides voters with a receipt about their vote. However, the scheme shows the receipt in an editable text field to the voter, *i.e.* if the voter intends to forward that receipt to the adversary, the adversary would not be convinced as the voter might manipulate the receipt before forwarding it. The adversary would merely be convinced about the receipt if he either controls the voting device, *i.e.* has the capability *voting device*, or observes the channel between a voter and her device, *i.e.* observes the reception of the return code. In both cases, the adversary could already violate vote secrecy in the original scheme.

Vote Integrity. The introduction of a cast-as-intended verification measure presented in Section 5.3 prevents a compromised voting device from altering votes in an undetectable manner. Consequently, in contrast to the original Polyas scheme, the capability *voting device* does not allow an adversary to violate vote integrity. Furthermore, the adversary controlling both the communication channel and being computationally unrestricted can

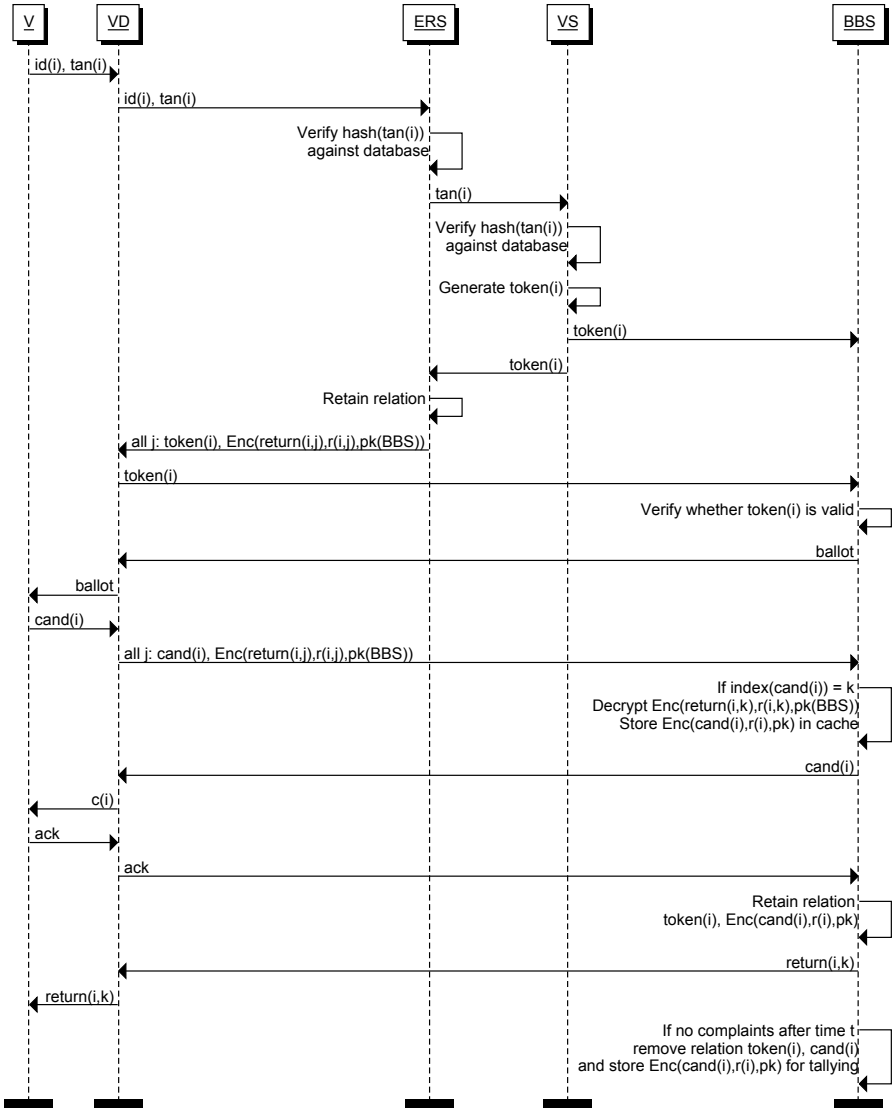

Figure 5.5: Voting phase of the extended Polyas voting scheme.

no longer undetectably alter votes. The only new possibility that the *ERS* obtains by its duty of generating return codes is to forward different codes to the *PS* and to the voter (throughout the voting phase). In that case, while the voter's vote might be transmitted correctly to the *BBS*, the match between the return code obtained from the *BBS* and the code list would not succeed. Yet, the *ERS* would not be able to manipulate the vote cast by the voter.

Eligibility. The authentication process is not affected by the improvement.

Fairness. In analogy to the argumentation line for vote secrecy, the extended Polyas scheme does not provide the adversary with new attack strategies with regard to fairness.

Data Access Protection. The extension does not provide any new component with voter data. Consequently, the qualitative security model remains unchanged with regard to data access protection.

Requirement	Qualitative Security Models	Impact
Vote Integrity	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} \cancel{VD_i})\right) \vee BBS \vee \cancel{(CR \wedge CCH)}$	$1 \leq l \leq n$
Eligibility	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\right) \vee (ERS \wedge VS) \vee (ERS \wedge PS)$	$1 \leq l \leq n$
Fairness	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} HCH_i)\right) \vee BBS \vee (CR \wedge CCH)$	$1 \leq l \leq n$
Vote Secrecy	$(ERS \wedge BBS) \vee (PS \wedge VS \wedge BBS) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} HCH_i)\right) \vee$	$1 \leq l \leq n$
Data Access Protection	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\right) \vee PS \vee ERS$	$1 \leq l \leq n$

Table 5.2: Qualitative security models of the extended Polyas Internet voting scheme. Struck through capabilities indicate that an adversary possessing those capabilities is no longer able of violating the respective security requirement because of the presented extension.

5.5. Comparison of the Qualitative Security Models of the Original and the Extended Scheme

The extended Polyas Internet voting scheme improves the original scheme by removing one assumption about adversarial capabilities. This extension is *not* accompanied by new assumptions with regard to any security requirement. Consequently, according to Definition 7, one can construct a simple identity mapping between both schemes, such that Pareto dominance of the extended scheme over the original scheme is given. The qualitative dominance of the extended scheme against the original scheme makes a quantitative comparison between both schemes obsolete. It might nevertheless be of interest to quantitatively evaluate both schemes within different election settings: If election officials consider decision criteria beyond security requirements, then election officials might consider the original Polyas scheme, *e.g.* because the use of return codes might be inappropriate for the target election. Then election officials might compare the satisfaction degree of the original and the extended scheme in their specific election setting, and take the relative security degrees

of both schemes as one out of several decision criteria. We therefore compare the security of both schemes quantitatively within five election settings.

5.6. Comparison of the Quantitative Security Models of the Original and the Extended Scheme

Election Settings. On the basis of the qualitative security models, the security of the original and the extended Polyas scheme are quantitatively assessed against the four probabilistic adversaries specified in Section 4.2.

Additionally, we construct a fifth adversary against which we expect the proposed extension to be irrelevant. The adversary corresponds to the adversary of election setting 1, except that the fifth adversary possesses the capability *VD* with a probability of 0. The probabilistic adversaries considered for the quantitative evaluation of the original and the extended Polyas Internet voting scheme are shown in Table 5.3. Due to its Pareto dominance, we expect the extended scheme to satisfy all security requirements in all election settings at least as good as the original scheme.

Referring to the GI 2011 election, we consider a number of $20,000$ eligible voters and $3,244$ expected voters[17].

Results. The results of the quantification process are provided in Tables 5.4, 5.5, 5.6 and 5.7, and are visualized in Figures 5.6, 5.7, 5.8, 5.9, 5.10. In addition to the satisfaction degrees, the tables contain the minimum and maximum theoretically possible satisfaction degrees for both schemes: a minimum (respectively maximum) satisfaction degree corresponds to the quantitative evaluation of qualitative security models with the largest (respectively smallest) probability value for all adversarial capabilities.

The obtained results confirm our expectation: The quantitative security evaluation results of the extended scheme are at least as high as the results of the original scheme. However, it turns out that the significance of the proposed extension varies with regard to the election settings.

Consider the results of the first election setting as baseline.

One can notice a significant increase in the difference between the satisfaction degrees of vote integrity once the adversary becomes stronger with regard to voting device corruption. This significance stems from the fact that an adversary controlling the voters' voting devices cannot undetectably violate vote integrity in the extension.

In contrast to the baseline setting, an adversary particularly strong with regard to service provider corruption impacts both schemes to approximately the same extent. This observation indicates that the proposed extension does not address vulnerabilities caused by service provider corruption. In fact, it can be noticed that the satisfaction degrees of

[17] Refer to `https://www.gi.de/wir-ueber-uns/unsere-mitglieder.html` and `https://www.gi.de/index.php?id=wahlen2011`

both schemes do not drop significantly, *i.e.* risks caused by compromised service providers are not the most prevalent risks for both voting schemes.

If the adversary increases his capabilities with regard to influencing voters, one can notice significant decreases in the satisfaction degrees of fairness, vote secrecy, and eligibility in both schemes. These decreases indicate that voters that are to some extent under adversarial control pose a serious security vulnerability to the scheme. On the other side, one can notice that the difference between both schemes with regard to vote integrity remains more or less unchanged in comparison to the baseline setting. This indicates that the proposed extension does not address vulnerabilities caused by voters that are under adversarial control.

Eventually, if the adversary does not have the capability to compromise voting devices, the quantitative difference between the original and the extended Polyas scheme vanishes. This observation is explained by the fact that the proposed extension targets specifically at this capability. Hence, the absence of this capability results in the fact that the difference in satisfaction degrees drops to 0.

5.7. Summary

With more than 2,2 millions cast online votes, the Polyas Internet voting scheme is one of the most established Internet voting schemes. Yet, the scheme unveils numerous shortcomings.

To address the risk of vote integrity violations caused by compromised voting devices, we reviewed existing technical solutions. Considering the constraints given by the Common Criteria certification, we presented an extension of the Polyas Internet voting scheme. By providing voters with code sheets, the ballot box server gains the possibility to confirm a vote by returning the respective return codes to the voter. Given the fact that the voting device only learns the return codes for the vote that has been received by the ballot box server, the voting device can only obtain the return codes that the voter expects by forwarding the voter's vote in an unaltered manner.

The qualitative security models show that the extended scheme Pareto dominates the original scheme. In the case of vote integrity, we were able to eliminate the need to trust the voting device, *without* imposing new assumptions on the adversary's capabilities.

The Pareto dominance of the extended Polyas scheme makes a quantitative security evaluation for the comparison of the extended and the original scheme obsolete. In all possible election settings, satisfaction degrees of the extended scheme are larger or equal than the respective satisfaction degrees of the original scheme. Yet, when taking into account decision criteria beyond legally-founded security requirements for Internet voting schemes, the security improvements might become one among several criteria. We therefore quantitatively evaluated both schemes within five election settings. The evaluation results show that the added value to the requirement vote integrity depends on the target

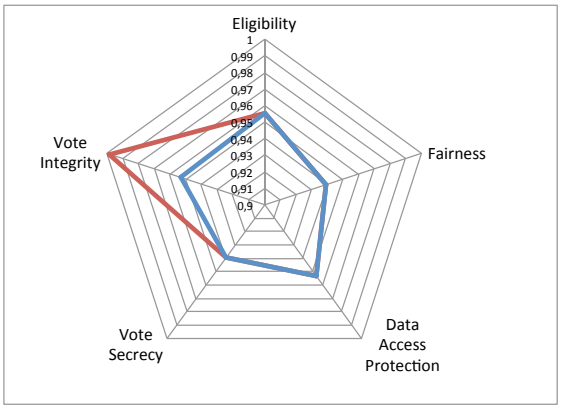

Figure 5.6: Polyas result: Election setting 1.

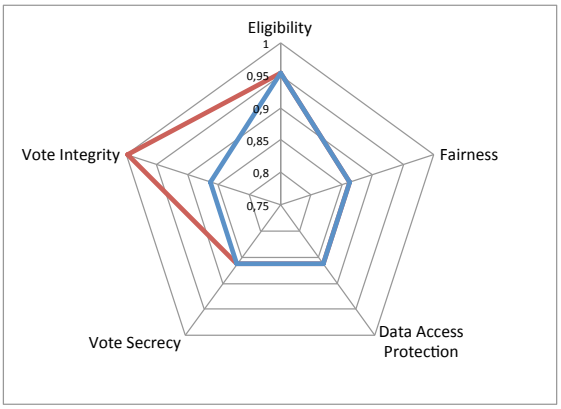

Figure 5.7: Polyas result: Election setting 2.

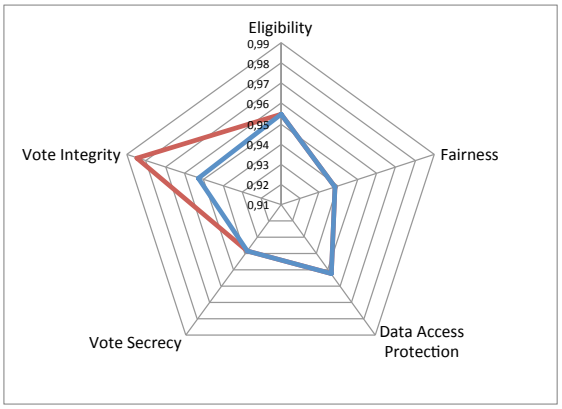

Figure 5.8: Polyas result: Election setting 3.

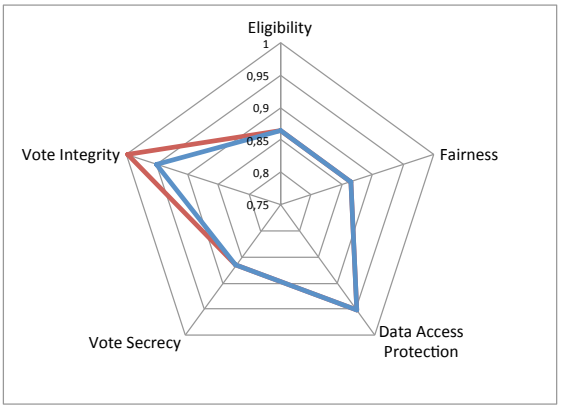

Figure 5.9: Polyas result: Election setting 4.

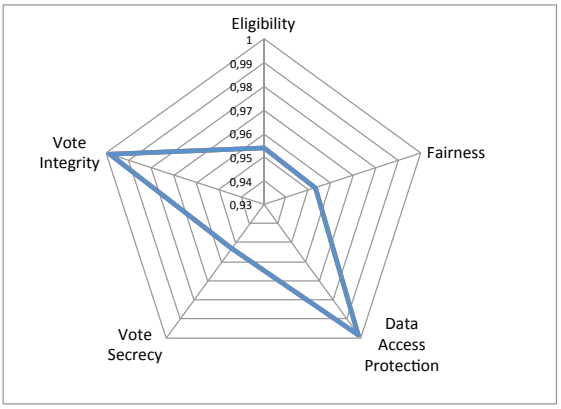

Figure 5.10: Polyas result: Election setting 5.

Election Setting	VD	ONSP	OFSP	VO	VI	HCH
E_1	U[0.01, 0.1]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_2	U[0.1, 0.2]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_3	U[0.01, 0.1]	U[0.01, 0.02]	U[0.001, 0.002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_4	U[0.01, 0.1]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.1, 0.2]	U[0.1, 0.2]	U[0.1, 0.2]
E_5	U[0, 0]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]

Table 5.3: Probabilistic adversaries considered for the quantitative evaluation of the original and extended Polyas scheme.

Requirement	Ori. Polyas SD	Ori. Polyas Min/Max SD	Ext. Polyas SD	Ext. Polyas Min/Max SD
Eligibility	0.955435709	[0.912297378316, 0.992843571063]	0.955435709	[0.912297378316, 0.992843571063]
Fairness	0.939280721	[0.907360601091, 0.991610302379]	0.939280721	[0.907360601091, 0.991610302379]
DA Protection	0.9532711	[0.912292977530, 0.992842627116]	0.9532711	[0.912292977530, 0.992842627116]
Vote Secrecy	0.939283299	[0.907363978273, 0.991610852262]	0.939283299	[0.907363978273, 0.991610852262]
Vote Integrity	0.953271423	[0.912293377019, 0.992842712867]	0.998502195	[0.998000000000, 0.999000000000]

Table 5.4: Results of the quantitative security evaluation of the original and extended Polyas scheme within election setting 1.

Requirement	Ori. Polyas SD	Ori. Polyas Min/Max SD	Ext. Polyas SD	Ext. Polyas Min/Max SD
Eligibility	0.953331026	[0.912297378316, 0.992843571063]	0.953331026	[0.912297378316, 0.992843571063]
Fairness	0.863296551	[0.817738026681, 0.912295378468]	0.863296551	[0.817738026681, 0.912295378468]
DA Protection	0.863308735	[0.817737487707, 0.912295178523]	0.863308735	[0.817737487707, 0.912295178523]
Vote Secrecy	0.863300222	[0.817743416421, 0.912297377915]	0.863300222	[0.817743416421, 0.912297377915]
Vote Integrity	0.851969229	[0.833560677883, 0.912295378468]	0.998487723	[0.998000000000, 0.999000000000]

Table 5.5: Results of the quantitative security evaluation of the original and extended Polyas scheme within election setting 2.

Requirement	Ori. Polyas SD	Ori. Polyas Min/Max SD	Ext. Polyas SD	Ext. Polyas Min/Max SD
Eligibility	0.954865328	[0.912297219964, 0.992843554076]	0.954865328	[0.912297219964, 0.992843554076]
Fairness	0.938335715	[0.907330145537, 0.990000000000]	0.938335715	[0.907330145537, 0.990000000000]
DA Protection	0.952400304	[0.912252141433, 0.989010000000]	0.952400304	[0.912252141433, 0.989010000000]
Vote Secrecy	0.93837185	[0.907363308118, 0.991610797764]	0.93837185	[0.907363308118, 0.991610797764]
Vote Integrity	0.952556556	[0.912256679453, 0.990000000000]	0.984888738	[0.980000000000, 0.990000000000]

Table 5.6: Results of the quantitative security evaluation of the original and extended Polyas scheme within election setting 3.

Requirement	Ori. Polyas SD	Ori. Polyas Min/Max SD	Ext. Polyas SD	Ext. Polyas Min/Max SD
Eligibility	0.86435605	[0.817743425062, 0.912297379517]	0.86435605	[0.817743425062, 0.912297379517]
Fairness	0.865989764	[0.817738026681, 0.912295378468]	0.865989764	[0.817738026681, 0.912295378468]
DA Protection	0.951977457	[0.912292977530, 0.992842627116]	0.951977457	[0.912292977530, 0.992842627116]
Vote Secrecy	0.865993359	[0.817743416421, 0.912297377915]	0.865993359	[0.817743416421, 0.912297377915]
Vote Integrity	0.951977703	[0.912293377019, 0.992842712867]	0.998505387	[0.998000000000, 0.999000000000]

Table 5.7: Results of the quantitative security evaluation of the original and extended Polyas scheme within election setting 4.

Requirement	Ori. Polyas SD	Ori. Polyas Min/Max SD	Ext. Polyas SD	Ext. Polyas Min/Max SD
Eligibility	0.95384898	[0.912297378316, 0.992843571063]	0.95384898	[0.912297378316, 0.992843571063]
Fairness	0.952939981	[0.912293377019, 0.992842712867]	0.952939981	[0.912293377019, 0.992842712867]
DA Protection	0.998346268	[0.997800400000, 0.998900100000]	0.998346268	[0.997800400000, 0.998900100000]
Vote Secrecy	0.95294243	[0.912297371911, 0.994012368095]	0.95294243	[0.912297371911, 0.994012368095]
Vote Integrity	0.998496099	[0.998000000000, 0.999000000000]	0.998496099	[0.998000000000, 0.999000000000]

Table 5.8: Results of the quantitative security evaluation of the original and extended Polyas scheme within election setting 5.

election setting. The higher the relative risk of voting device corruption (in relation to other adversarial capabilities), the higher is the relevance of the proposed extension.

Chapter 6

The Estonian Internet Voting Scheme as Applied for the Parliamentary Elections 2015

In 2005, Estonia became the first country world-wide to introduce Internet voting for legally-binding elections [Kal09]. Since that time, Estonia has conducted a number of legally-binding elections. It turns out that the Internet voting option is getting more and more popular among Estonian citizens. For the European parliamentary election in 2014 and the Estonian parliamentary elections in 2015, more than 30% of all participating voters have cast their vote over the Internet[18]. The Estonian Internet voting scheme has been extensively described in public documents and academic literature, *e.g.* [MM06, Tre07, SFD+14].

The first part of this chapter provides an overview about the Estonian Internet voting scheme as applied for the Estonian parliamentary elections 2015. We thereafter qualitatively evaluate the security of the Estonian Internet voting scheme. Subsequently, we address several shortcomings of the original scheme by the construction of an extended scheme. The security of the extended scheme is qualitatively evaluated. We compare the qualitative security models of the original and extended scheme. Subsequently, the security of both schemes is quantitatively compared within different election settings. The chapter is concluded with a summary section.

Parts of this chapter have been published at the *Eighth International Conference on Availability, Reliability and Security* (ARES2013) [3].

6.1. Original Scheme

Before presenting the protocol underlying the Estonian Internet voting scheme, we describe the involved components.

[18]Refer to `http://www.vvk.ee/voting-methods-in-estonia/engindex/statistics`

6.1.1. Components

The Estonian Internet voting scheme is composed of the following components.

National Electoral Committee (NEC). The national electoral committee consists of seven members. The members of the *NEC* are particularly in charge of activating the hardware security module for vote tallying. The members of the *NEC* operate in an offline manner.

Vote Forwarding Server (VFS). The *VFS* is the Estonian Internet voting scheme's interface to the voters. The *VFS* provides voters with digital ballots and forwards filled ballots to the vote storage server. The *VFS* is the only server in the Estonian Internet voting scheme that operates in an online manner.

Vote Storage Server (VSS). The *VSS* collects filled ballots and anonymizes votes for vote tallying. The *VSS* operates in an offline manner.

Vote Counting Server (VCS). The *VCS* receives anonymized votes and by the use of the hardware security module runs the tallying process. The *VCS* operates in an offline manner.

Log Server (LS). The *LS* is a server that is connected to the *VFS* and the *VSS* and constantly receives log information from those servers. The *LS* operates in an offline manner.

Hardware Security Module (HSM1/HSM2). The $HSM1$ is embedded into the vote counting server and is in charge of generating the election key pair and decrypting encrypted votes after the module has been activated by the *NEC*. The $HSM1$'s operation in terms of vote tallying can only be initiated if at least four out of the seven *NEC* members provide their authentication material; these are PIN-protected keycards. In addition to the $HSM1$, there exists a backup $HSM2$. $HSM2$ stores the same information and has the same functionality as $HSM2$. Both, $HSM1$ and $HSM2$ operate in an offline manner.

Voting Device (VD1 / VD2). Each voter has two voting devices at her disposal. The voter uses voting device $VD1$ to fill and encode her ballot and $VD2$ to audit the encoding process conducted by $VD1$.

6.1.2. Ballot of the Estonian Parliamentary Elections 2015

The Estonian parliamentary (*Riigikogu*) election is held every four years within twelve electoral districts. Each district has its own list of candidates. The district with the fewest candidates has 49 candidates, the district with most candidates has 115 candidates.

Each voter has the possibility to select *one* candidate from the ballot of her electoral district. There is no option to cast an invalid vote.

6.1.3. Protocol Description

The following description builds upon upon the general system description by the Estonian National Election Committee [Com10], Heiberg *et al.* [HLV12], Heiberg and Willemson [HW14], and Springall *et al.* [SFD+14] which takes into account smartphone verifiability measures introduced into the Internet voting scheme in 2013. In analogy to the Polyas scheme, we consider components of the scheme to operate in independent manner. To propagate this aspect to the system layer, the components have to be implemented and hosted by independent providers.

Setup Phase

In advance to the election, all involved service providers generate SSL/TLS and signature keys and publish the respective public keys. Furthermore, an asymmetric election key pair (pk, sk) is generated within $HSM1$. The public election key is implemented into the voting application. The voting application is subsequently signed by the *NEC* and distributed to the voters. The protocol steps of the Estonian setup phase are depicted in Figure 6.1.

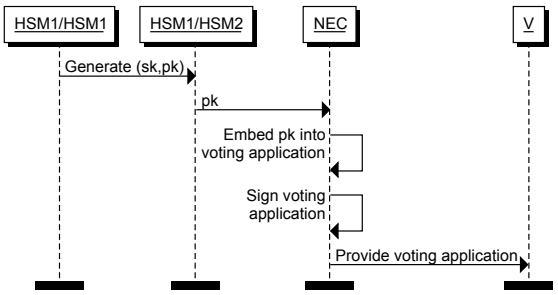

Figure 6.1: Setup phase of the original Estonian Internet voting scheme.

Voting Phase

Before starting the voting process, the voter visits the election website from which the voting application can be downloaded. To start the vote casting process, the voter launches her voting application which establishes an encrypted and authenticated connection towards the *VFS*. All knowledge the *VFS* obtains throughout the voting phase is synchronized with the *LS*. The voter authenticates herself towards the *VFS* by the use of her eID card (with her authentication PIN). The *VFS* determines whether the voter is eligible to vote by looking up the voter's identification number in the actual voter list and in which district the voter resides to provide the respective candidate list. The *VFS* consults the

VSS to determine whether the voter has already cast a vote. Analogously to *VFS*, all knowledge the *VSS* obtains throughout the voting phase is synchronized with the *LS*. The *VFS* returns the respective candidate list and the information whether she has already voted or not to the voter's voting application. Once the voter obtained the list of candidates from the *VFS*, she makes her selection and fills the ballot accordingly. The voting application pads the voting option with a randomization factor r and encrypts the ballot with the public election key stored in the voting application. The voter signs her encrypted ballot with her signature key (with her signature PIN). Thereafter, the signed and encrypted ballot is sent to the *VFS*. The *VFS* checks whether the signature relates to the voter who authenticated in first place. If so, the *VFS* forwards the ballot to the *VSS*. The *VSS* requests a validity certificate for the signed vote from the external validity confirmation service. That service issues validity certificates for voter signatures on encrypted ballots[19]. The *VSS* stores the signed encrypted vote together with a time stamp for the purpose of vote tallying. Furthermore, the *VSS* stores the validity certificate and assigns a value x to the ballot. The value x is returned to the voter's client and the voting application presents the randomization factor r and value x in form of a QR code for the purpose of verification. After receiving the vote confirmation and the value x, the voter can optionally conduct a vote verification step, as outlined in Section 4.1 (Independent Verification Devices). For the sake of preventing adversaries from influencing voters, voters can arbitrarily often update their Internet vote in the remote voting phase. The protocol steps of the Estonian voting phase are depicted in Figure 6.2.

Tallying Phase

At the end of the election, the signatures of valid encrypted ballots (last Internet votes that have been cast by eligible voters) are removed and the unpersonalized ballots are transmitted (via a burned DVD) to the *VCS*. The *HSM*1 within the *VCS* is activated by at least 4 out of 7 *NEC* members upon which the *HSM*1 decrypts the anonymized ballots. Eventually, the election result is announced. An overview about the tallying phase is provided in Figure 6.3.

6.2. Qualitative Security Models of the Original Scheme

On the foundation of scientific literature [MV11, HLV12, HW14, SFD+14], particularly focusing on the security analysis by Springall *et al.* [SFD+14] and informal protocol analysis, we take the role of system analysts and determine qualitative security models of the Estonian Internet voting scheme. In the remainder of this section, we outline which capabilities allow an adversary to cause impact on security requirements. An overview of the result is given in Table 6.1. Recall that we make the general assumption that

[19]The validity confirmation service is not considered a component of the Internet voting scheme as the service is widely used for a variety of Estonian eID services.

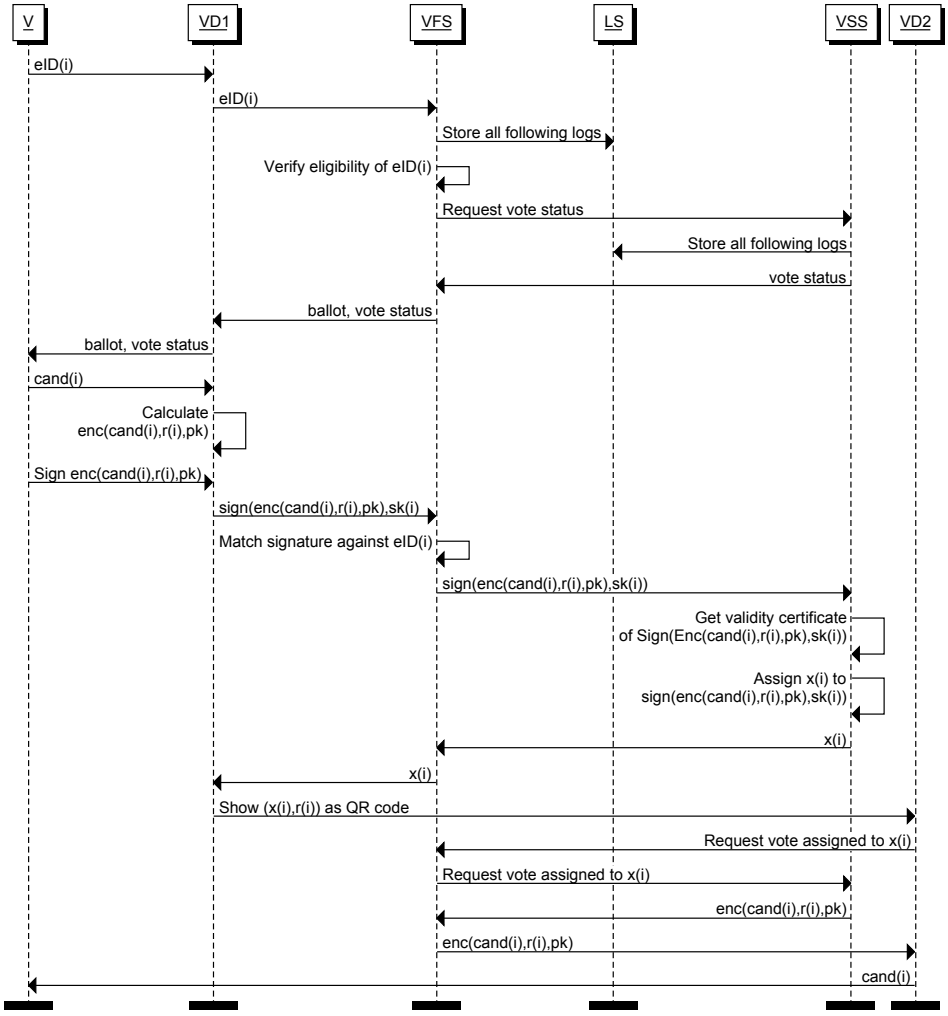

Figure 6.2: Voting phase of the original Estonian Internet voting scheme.

anything that can be verified is verified. Furthermore, the log server's purpose is essentially to monitor the election remotely and to detect external attacks launched at run-time. Recall that adversarial capabilities represent static corruption, *i.e.* either an adversarial capability is given or it is not given. Consequently, if either *VFS* or *VSS* are compromised, external log files are adapted accordingly by the compromised components. Hence, in the case of static adversaries, there is no benefit of the *LS*. We therefore do not consider *LS* as dedicated component in the qualitative security models.

Vote Secrecy. The Estonian Internet voting scheme provides vote updating as measure to counter conscious observation attacks. However, this measure is irrelevant if voters are unconscious about being under observation, *i.e.* in the case their voting device is compromised. Madise and Vinkel [MV11] show that in the parliamentary elections 2011,

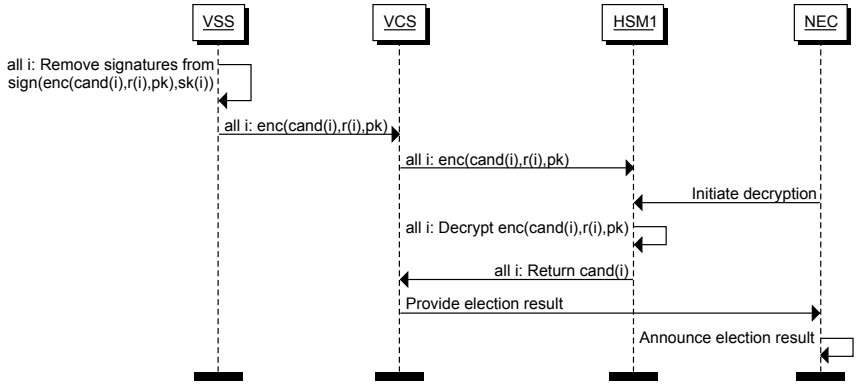

Figure 6.3: Tallying phase of the original Estonian Internet voting scheme.

only 3.1% of cast Internet votes were not tallied because of vote updating. Corrupting the following components allows the adversary to violate vote secrecy.

Voting Device. In the Estonian Internet voting scheme, the voter selects her preferred voting option within the client application on her device. Compromised voting devices might consequently store this selection and relate it to the voter's identity, thereby violating vote secrecy of the vote cast over that device. Given the fact that throughout the parliamentary elections 2011, $3,1\%$ of voters updated their votes, we assume that corrupting the voting device does only work up to an impact level of 97%[20].

Vote Forwarding Server, Vote Counting Server, Hardware Security Modules, and the NEC. There are components that know the relation between a ciphertext and the identity of the voter who cast that ciphertext. Furthermore, there are components that know the relation between a ciphertext and the respective plaintext vote. If at least one component of each group maliciously collaborate by combining their respective knowledge, the conspiracy is able to violate vote secrecy of all votes. Throughout the voting phase, the *VFS* learns for all participating voters the former relation. The $HSM1$, the $HSM2$, the *VCS*, and the threshold of *NEC* members (might) know the latter relation. Hence, the conspiracy between *VFS* and either the $HSM1$, the $HSM2$, the *VCS*, and the threshold of *NEC* members are able to violate vote secrecy of all votes.

Computationally unrestricted and Communication Channel. If an adversary is capable of controlling the communication channel between the voting device and the *VFS* and is capable of breaking cryptographic primitives, then the adversary can break vote secrecy, because encrypted votes are signed by the voters who cast them.

[20] In the future, the security evaluation framework can be extended by parametrizing the percentage of expected vote updates. In that case, election officials the estimated percentage of expected vote updates might be provided by election officials as part of the election setting specification.

Vote Integrity. Corrupting the following components allows the adversary to violate vote integrity within the Estonian Internet voting scheme.

Voting Device. The Estonian Internet voting scheme has recently been improved towards vote integrity with regard to compromised voting devices by providing a verification mechanism [HW14]. Yet, as shown by Springall *et al.* [SFD+14], compromised voting devices might launch Ghost click attacks [SFD+14], thereby replacing voters' cast and potentially verified votes. In order to launch such an attack, the compromised device would re-use the voter's PIN once her eID is entered into the card reader. Given the fact that Estonian citizens tend to use their eIDs frequently for different services [Rep14] and smart card readers without PIN pad are mostly used by Estonian citizens[21], such an attack could be successfully launched on a large-scale. Yet, we exclude those voters that override their Internet vote and those voters that would not use other services. We therefore assume the maximum impact by 80% of all votes[22]. In spite of the good intention to improve vote integrity, the implemented verification mechanism turns out to be of low effectivity.

Vote Forwarding Server and Vote Storage Server. By counting the number of cast votes, the *VFS* controls the *VSS*'s behaviour. If the *VSS* illegitimately drops valid ballots, a discrepancy in the number of processed ballots between the *VFS* and the *VSS* would be detected by the *NEC*. However, if both components, the *VFS* and the *VSS*, maliciously collaborate, they might drop valid ballots undetectably, thereby violating integrity of all cast votes.

Vote Storage Server and NEC. In the tallying phase, it is the duty of the *VSS* to separate signatures from received votes and output only anonymized votes for the tallying process. If *VSS* would alter votes after the signature has been stripped off, *NEC* would discover this misbehaviour by the log files generated by the *VFS* and *VSS*. The log files generated by the *VFS* and *VSS* are not publicly available, but remain under control of the *NEC* [HPW15]. Hence, if the *VSS* maliciously collaborates with the *NEC*, the conspiracy might alter votes undetectably.

Vote Counting Server. Throughout the tallying phase, the *VCS* might store votes differently than it obtained these votes from the *HSM*1. Thereby, the *VCS* is capable of violating vote integrity of all stored votes.

Hardware Security Module 1. The *HSM*1 might forward votes differently than decrypted, thereby violating vote integrity of the stored votes.

Computationally unrestricted and Communication Channel. If the adversary is capable of controlling the communication channel between the voting device and the *VFS* and is capable of breaking cryptographic primitives, then the adversary can drop voters' votes, and replace them by other forged votes.

[21] Refer to the recommendations under `http://www.id.ee/index.php?id=35612`

[22] Analogously to the percentage of vote updates, the expected number of voters who override their Internet vote or do not use other Internet services might be parametrized and provided by the election official.

Eligibility. We assume that all citizens that use their eIDs for authentication and digital signatures in Internet based services also cast their votes via the Internet voting channel. Hence, if a compromised voting device learns the voter's PIN and illegitimately casts a vote, this corresponds to a violation of vote integrity rather than eligibility. In spite of the strong authentication means and the external validity confirmation service, log files to verify the eligibility of cast votes remain under control of the *NEC*. Consequently, eligibility is not verifiable by the general public. Hence, corrupting the following components allows the adversary to violate eligibility within the Estonian Internet voting scheme.

Vote Storage Server and the NEC. Analogously to the vote integrity case, if the *VSS* is compromised and adds illegitimate votes and *NEC* is corrupt, the conspiracy could undetectably add illegitimate votes, thereby violating eligibility.

Computationally unrestricted and Communication Channel. If the adversary is capable of controlling the communication channel between the voting device and the *VFS* and is capable of breaking cryptographic primitives, then he can determine which voters cast votes. The adversary can subsequently cast forged votes on behalf of abstaining voters.

Fairness. Attack strategies against fairness closely resemble strategies against vote secrecy. Analogously to the vote secrecy case, we consider that 3.1% of all cast votes are updated. Consequently, only votes not being updated may be revealed reliably by the adversary before the end of the voting phase. Corrupting the following components allows the adversary to violate fairness within the Estonian Internet voting scheme.

Voting Device. Analogously to the vote secrecy case, compromised voting devices might leak the voter's selection to the adversary, thereby violating the fairness requirement. One compromised device might thereby reveal the selection cast over that device.

Hardware Security Module 1/2. In case of corruption, both $HSM1$ and $HSM2$ are able to decrypt all cast votes before the voting phase ended, thereby violating fairness.

The NEC. Rather than breaking into hardware security modules, the adversary might corrupt a threshold of election officials to launch the hardware security module to decrypt all cast votes before the voting phase ended, thereby violating fairness.

Computationally unrestricted and Communication Channel. Analogously to the vote secrecy, if the adversary is capable of controlling the communication channel between the voting device and the *VFS* and is capable of breaking cryptographic primitives, he can decrypt encrypted votes, thereby violating fairness.

Data Access Protection. As part of the protocol, the following components obtain voter data. Hence, their corruption can result in violation of data access protection.

Voting Device. Because of the fact that voting devices are generally used for a number of purposes, voting devices know the voters' identities. A compromised voting device might consequently forward a voter's identity to the adversary.

Vote Forwarding Server. Due to the double envelope method implemented by the Estonian Internet voting scheme, the *VFS* receives signed ballots from the voters. A compromised *VFS* might consequently forward identities of voters to the adversary.

Vote Storage Server. Analogously to the *VFS*, the *VSS* receives and additionally stores signed ballots throughout the voting phase. Corrupting the *VSS* consequently allows an adversary to violate data access protection.

Computationally unrestricted and Communication Channel. By controlling the communication channel between the voting device and the *VFS*, and additionally breaking cryptographic primitives, the adversary can determine the identities of voters who cast an Internet vote.

Requirement	Qualitative Security Models	Impact
Vote Integrity	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD1_i)\right)\vee$ $(VFS\wedge VSS)\vee$ $(VSS\wedge(7 \text{ out of } NEC))$ $VCS\vee HSM1\vee(CR\wedge CCH)$	$0\leq l\leq\frac{80}{100}$
	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD1_i\wedge VD2_i)\right)\vee$ $(VFS\wedge VSS)\vee$ $(VSS\wedge(7 \text{ out of } NEC))$ $VCS\vee HSM1\vee(CR\wedge CCH)$	$\frac{80}{100}<l\leq 1$
Eligibility	$(VSS\wedge(7 \text{ out of } NEC))\vee(CR\wedge CCH)$	$0\leq l\leq 1$
Fairness	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD1_i)\right)\vee$ $HSM1\vee HSM2\vee(4 \text{ out of } NEC)\vee(CR\wedge CCH)$	$0\leq l\leq\frac{97}{100}$
Vote Secrecy	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD1_i)\right)\vee$ $(VFS\wedge(HSM1\vee HSM2\vee VCS\vee$ $(4 \text{ out of } NEC)))\vee(CR\wedge CCH)$	$0\leq l\leq\frac{97}{100}$
	$(VFS\wedge(HSM1\vee HSM2\vee VCS\vee$ $(4 \text{ out of } NEC)))\vee(CR\wedge CCH)$	$\frac{97}{100}<l\leq 1$
Data Access Protection	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD1_i)\right)\vee$ $VFS\vee VSS\vee(CR\wedge CCH)$	$1\leq l\leq n$

Table 6.1: Qualitative security models of the original Estonian Internet voting scheme.

Discussion

It turns out that the Estonian Internet voting scheme reveals significant vulnerabilities with regard to compromised voting devices. In fact, the security requirements vote in-

tegrity, fairness, vote secrecy, and data access protection can be violated by compromised voting devices.

As outlined earlier, this assumption is inadequate given the high infection rates of computers worldwide [Pan15]. Similar to the Polyas scheme, we extend the Estonian scheme by means of code voting. While the adaptations of the Polyas scheme were constrained by the current Common Criteria certification of the scheme, the Estonian scheme is adapted to a larger extent. In addition to return codes, we incorporate voting codes to improve the scheme with regard to legally-founded security requirements beyond vote integrity. To prevent single components from violating specific security requirements due to the incorporation of voting codes, we adapt the scheme further by rigorously separating duties between different components.

6.3. Proposed Extensions

Before diving into the details of our extension, we review related works on the Estonian Internet voting scheme and related works on code voting based schemes.

6.3.1. Related Work

The only efforts made to improve the Estonian scheme in the presence of compromised voting devices has been presented and explained by Heiberg and Willemson [HW14]. To counter vote integrity violations, the extension incorporates independent verification devices. As discussed earlier, given the facts that a high percentage of voters use their eID cards for other purposes and that vote updating is possible, malware on the voting device can circumvent the verification mechanism [SFD+14].

Securely voting over untrustworthy platforms (voting devices) was initially addressed by Chaum's SureVote [Cha01], the first code voting scheme. Numerous code voting schemes have been proposed later on, *e.g.* [JFR13], [JRF09], [JR07a, JR07b, JRF10], [Hel09, HS07, HSS08] and [ZCC+13]. The schemes in [Cha01], [JFR13], [JRF09], and [ZCC+13] assume the voter to be honest in order to ensure vote secrecy. Other extensions of code voting, [JR07a, JR07b, JRF10] assume a trustworthy voting- and voter-specific smart card for vote secrecy and integrity. Within all of these schemes, one component (either voter or smart card as instantiation of a voting device) can violate vote secrecy or vote integrity.

6.3.2. Components

In contrast to the original Estonian scheme, the extended scheme does not incorporate the vote counting server *VCS*, both hardware security modules $HSM1$ and $HSM2$. In addition, voters only need one voting device *VD*, rather than two. On the other side, the revised scheme incorporates three new components, namely a distribution authority *DA*, and two voting authorities the $VA1$ and the $VA2$. While several components are

maintained, the roles of these components have changed such that we provide an overview about all components in the following.

National Electoral Committee (NEC). The national electoral committee is composed out of seven members. Committee members are involved in the setup phase, in particular in generating a threshold election key pair (pk, sk). Each committee member possesses a share of the secret key. Committee members are also involved in the tallying phase. The *NEC* operates in an offline manner.

Distribution Authority (DA). The *DA* is involved in the setup phase; together with the *NEC*, it anonymizes, audits and distributes code sheets. Thus, both know the election register. DA operates in an offline manner.

Vote Forwarding Server (VFS). The *VFS*, in the setup phase, generates the code sheet parts containing the permuted list of candidates. The *VFS* is also involved in the voting phase and knows the election register. VFS operates in an online manner.

Voting Authority (VA1). The *VA*1, in the setup phase, generates codes. The *VA*1 is also involved in the voting phase. Furthermore the *VA*1 holds a signing key. The *VA*1 operates in an offline manner.

Voting Authority (VA2). The *VA*2 has a similar functionality as the *VA*1. The *VA*2 operates in an offline manner.

Vote Storage Server (VSS). The *VSS* is involved in all phases. Any component has read access, all service providers (except *DA*) have write access. All data published on the *VSS* are signed by the sending service providers[23]. The *VSS* provides different sectors for all phases. The *VSS* operates in an online manner.

Voting Device (VD). As opposed to the original Estonian scheme, in the extended scheme, each voter has one voting device at her disposal over which she casts her vote. The second voting device is no longer necessary as cast-as-intended verifiability is provided by the code voting approach.

6.3.3. Code Sheets in the Extended Estonian Scheme

The code sheets used in the proposed extension consist of three parts (*i.e.* three different pieces of paper), two parts containing codes and one part containing a permuted list of

[23] Sending service providers compute one signature over all data in one protocol step. Note, in Figures 6.7 and 6.11 the signatures are not illustrated.

candidates. Each code sheet part is generated by a different service provider. The three code sheet parts are linked by their index to one code sheet.

An example of one part of the code sheet containing codes is depicted in Figure 6.4. This part with accompanying index i is generated by the $VA1$, whose identity is also indicated, next to the acknowledgment code. $Code_{VA1,i,1} \ldots, Code_{VA1,i,n}$ denote n random, unique codes and $Ack_{VA1,i}$ denotes a random, unique acknowledgment code. Similarly, the $VA2$ generates the second part of the code sheet containing codes for index i.

i
$Code_{VA1,i,1}$
$\vdots$
$Code_{VA1,i,n}$
$Ack_{VA1,i}$

Figure 6.4: Code sheet part generated by the $VA1$ with index i.

The third part of the code sheet with index i is generated by the VFS and consists of the list of n candidates, randomized according to a secret permutation ϕ_i. Recall that in the case of the Estonian parliamentary elections 2015, the list contains between 49 and 115 candidates. The code sheet part containing the candidates is shown in Figure 6.5 and a complete code sheet is illustrated in Figure 6.6.

i
$\phi_i(Candidate_1)$
$\vdots$
$\phi_i(Candidate_n)$
–

Figure 6.5: Code sheet part with index i generated by the VFS.

i
$Code_{VA1,i,1}$
$\vdots$
$Code_{VA1,i,n}$
A: $Ack_{VA1,i}$

i
$Code_{VA2,i,1}$
$\vdots$
$Code_{VA2,i,n}$
B: $Ack_{VA2,i}$

i
$\phi_i(Candidate_1)$
$\vdots$
$\phi_i(Candidate_n)$
–

Figure 6.6: Code sheet in extended Estonian scheme.

For a code sheet with index i, the voting code for the candidate in the p-th position is the concatenation of the corresponding codes in the p-th position:

$$Code_{i,p} = Code_{VA1,i,p} \parallel Code_{VA2,i,p}$$

Accordingly, the voting acknowledgment code of this code sheet is the concatenation of the acknowledgment codes:

$$Ack_i = Ack_{VA1,i} \parallel Ack_{VA2,i}$$

6.3.4. Revised Protocol Description

Analogously to the previously presented Internet voting schemes, we present the protocol underlying the actual scheme.

Setup Phase

The setup phase consists of key generation as well as generating, committing on, auditing, anonymizing and distributing code sheets.

Generating Keys. All involved service providers generate SSL/TLS and signature keys and publish the respective public keys. The *NEC* generates a threshold election key pair (*pk*, *sk*) in a distributed manner.

Generating Code Sheets. The *VFS* generates the part of each code sheet containing the candidates (refer to Figure 6.5): It randomizes the canonical order of the candidate list for each code sheet according to a secret permutation and prints the index and the randomized candidate list on a sheet of paper (refer to Figure 6.5). The *VFS* inserts its sheets of paper into privacy-protected sealed envelopes. The corresponding indexes are printed on the envelopes and sent to the *DA*. The *VA*1 and the *VA*2 independently generate random, unique codes for each candidate and each code sheet. They also independently generate random unique acknowledgment codes for each code sheet. Note that the acknowledgment codes must not match codes for candidates. The *VA*1 and the *VA*2 independently print this information on a sheet of paper (refer to Figure 6.4). The *VA*1 and the *VA*2 also insert their sheets of paper into privacy-protected sealed envelopes, print the corresponding indexes on the envelopes and send them to the *DA*. Note that more code sheets than eligible voters must be generated to make the auditing of code sheets possible.

Committing on Code Sheets. After generating the code sheet parts, the *VFS*, the *VA*1 and the *VA*2 commit on their respective parts: Committing is done by encrypting corresponding parts with the public election key *pk* and publishing the encryptions under the accompanying index in the setup phase sector of the *VSS*, see Figure 6.7. Note that committing is needed in order to detect a corrupt *VFS*, *VA*1, and *VA*2 distributing invalid code sheets.

Auditing Code Sheets. Afterwards, the *DA* and the *NEC* start with the auditing process, shown in Figure 6.8: The *NEC* randomly selects code sheets to be audited. The corresponding data for each code sheet to be audited is downloaded from the setup phase sector of the *VSS*. The downloaded data is decrypted by a threshold set of the *NEC*.

Vote Storage Server Setup Phase Sector
$\vdots$
i
$\{\phi_i(Candidate_1)\}_{pk} \ldots \phi_i(Candidate_n)\}_{pk}$
$\{Code_1 \ldots Code_n; Ack_{\mathrm{VA1}}\}_{pk}$
$\{Code_1 \ldots Code_n; Ack_{\mathrm{VA2}}\}_{pk}$
$\vdots$

Figure 6.7: Content of the VSS at the end of the setup phase in the extended Estonian scheme.

The decrypted data is matched against the content of the corresponding envelopes. The audited code sheets are then discarded. Note, this process can be observed by the general public, *e.g.* by video-streaming the process over the Internet.

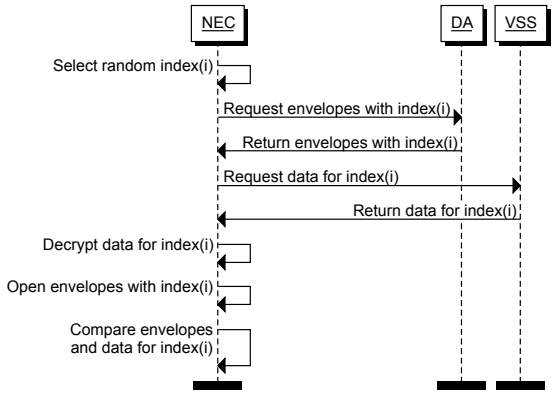

Figure 6.8: Auditing process of the extended Estonian Internet voting scheme.

Anonymizing and Distributing Code Sheets. After the auditing process, the *DA* in cooperation with the *NEC* anonymize and distribute the remaining envelopes to eligible voters, shown in Figure 6.9: All envelopes sharing the same index are placed into indistinguishable envelopes. These are put into a box and shuffled. After permuting, the *DA* and the *NEC* take the anonymized neutral envelopes out of the box, print voters' addresses on the envelopes and send them to the corresponding addresses.

Voting

The voter receives an envelope and checks that it contains the three code sheet parts, that the three code sheet parts are in privacy-protected sealed envelopes, and that the envelopes share the same index. The voter opens the three envelopes and combines the three code sheet parts in an order that is publicly known.

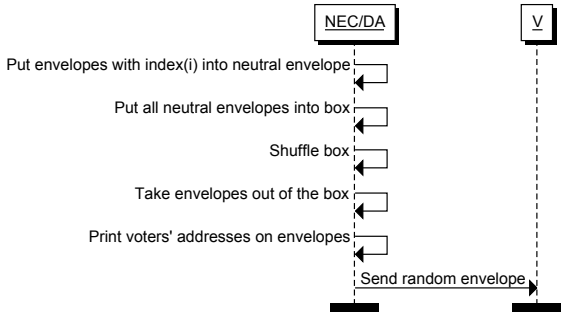

Figure 6.9: Anonymization and distribution process of the extended Estonian Internet voting scheme.

The vote casting process is shown in Figure 6.10. In order to vote, the voter authenticates herself by means of her eID to the voting website, which is hosted by the *VFS*. The *VFS* verifies that the voter is eligible to vote. If so, the *VFS* allows the voter to cast her vote. To cast a vote, the voter enters the voting code matching the candidate of her choice on the voting website. Recall that in the case of the Estonian parliamentary elections 2015, voters are allowed to cast one voter for one candidate from the candidate list, *i.e.* each voter enters exactly one voting code[24]. The voter signs the voting code with her eID card and transmits the signed code to the *VFS*. The *VFS* requests a validity certificate for the signed vote from the external validity confirmation service. The *VFS* stores the signature, subsequently removes the signature from the voting code, and forwards the first part of the voting code to the *VA*1 and the second part to the *VA*2. First, the *VA*1 and the *VA*2 deduce the index and the acknowledgment code of the code sheet (based on the received code) and the corresponding position of the code. The *VA*1 and the *VA*2 cross-check that they obtained codes of the same index and the same position. In case the code is invalid or a mismatch is detected, the *VA*1 and the *VA*2 inform the *VFS* that informs the voter. If the check is positive, they request and obtain the encryption of the candidate for the index and the position from the *VSS* (refer to Figure 6.7, first row after the index i). The *VA*1 and the *VA*2 independently re-encrypt the received ciphertext to $\{\phi_i(Candidate_p)\}_{pk}^{r_1}$ and $\{\phi_i(Candidate_p)\}_{pk}^{r_2}$. After this, they send the re-encrypted ciphertexts to the *VSS*. The *VSS* publishes the received data and sends a confirmation to the *VA*1 and the *VA*2. The *VA*1 and the *VA*2 verify that the respective data has indeed been published by the *VSS*. Figure 6.11 illustrates the information on the *VSS*. After having received the confirmation, the *VA*1 and the *VA*2 store and/or update the request by the *VFS*, the number of voters for which votes have been cast, and forward the previously deduced acknowledgment codes to the *VFS*. The *VFS* concatenates these codes into the voting acknowledgment code, which it sends to the voter.

[24]It shall be emphasized that Estonian voters have to enter a candidate number if they cast their vote by paper ballot.

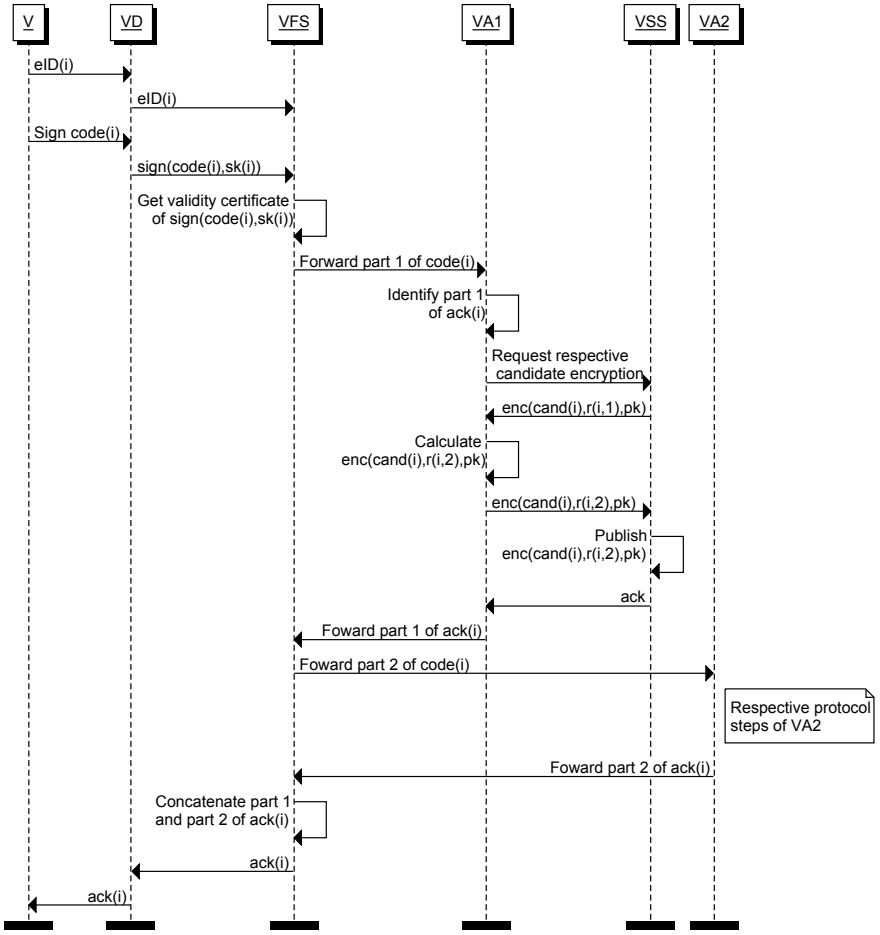

Figure 6.10: Voting phase of the extended Estonian Internet voting scheme.

Tallying

After the voting phase, each row of the *VSS* corresponds to a successfully cast vote (refer to Figure 6.11). The tallying process is shown in Figure 6.12. Before the process starts, the *VFS* sends the total number of voters who have cast a vote to the *VSS*. Analogously, the *VA*1 and the *VA*2 send the number of votes for which a re-encryption has been generated and published to the *VSS*. The general public can check that these numbers match the number of rows on the *VSS*. The committee members request the re-encrypted ciphertexts and the *VSS* sends back the data re-encrypted by the *VA*1 and the *VA*2, corresponding to column 1 and column 2 of the *VSS*'s voting phase sector. The *NEC* sums up the content of each individual column homomorphically. The encrypted sums are then decrypted by a threshold set of the *NEC*. The NEC compares the decrypted sums, and if they match, the *election result* is declared to be the matching sum. Finally, the committee members publish the ZKPs for correct decryption and the *election result* on the *VSS*.

Vote Storage Server Voting Phase Sector	
Column 1	Column 2
⋮	
$\{\phi_i(Candidate_p)\}_{pk}^{r_1}$	$\{\phi_i(Candidate_p)\}_{pk}^{r_2}$
⋮	

Figure 6.11: Content on VSS during the voting phase in the extended Estonian scheme.

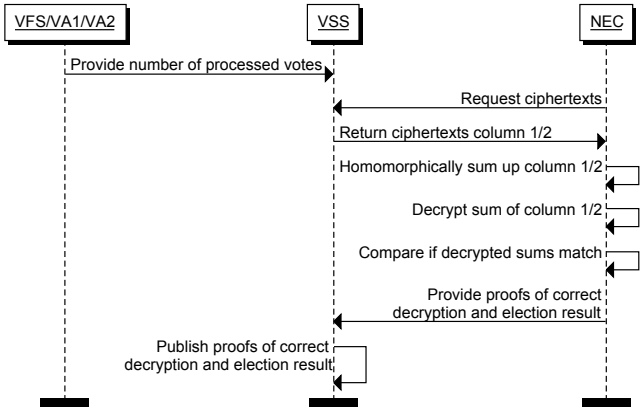

Figure 6.12: Tallying phase of the extended Estonian Internet voting scheme.

6.4. Qualitative Security Models of the Extended Scheme

In analogy to the original Estonian scheme, we take the role of system analysts and determine qualitative security models of the revised scheme. In the remainder of this section, we outline which capabilities allow an adversary to cause impact on security requirements. An overview of the result is given in Table 6.2. Recall that we make the general assumption that anything that can be verified is verified.

Vote Secrecy. In analogy to Section 6.2, we assume that throughout an election 3.1% of all votes are update votes and therefore not tallied. Corrupting the following components allows the adversary to violate vote secrecy.

Voter Output, and Voting Device, or Vote Forwarding Server, or one Voting Authority. In advance to the election, the voter receives her voting material in terms of code sheets. Code sheets capture in an un-encoded manner the relation between voting options and voting codes. If a voter forwards these code sheets to the adversary, the adversary might obtain a proof about the voter's vote if controls a component receiving at least half of the voter's voting codes. The components that are able to forward parts of the voter's voting code to the adversary are *VD*, the *VFS*, the *VA*1, and the *VA*2. Given the fact that voters might update their votes from different devices, breaking vote secrecy by obtaining the voter's

code from the voter's device *VD* might only work for those votes that are not updated. As opposed to *VD*, the service providers, namely the *VFS*, the *VA*1, and the *VA*2, are able to provide the adversary with the final voting code cast for each code sheet, thereby violating vote secrecy of all cast votes.

Vote Forwarding Server and one Voting Authority. Throughout the voting phase, the *VFS* learns the voter's real identity due to the use of strong authentication means. Additionally, the *VFS* learns the voter's cast voting code. Because the *VFS* published the encrypted voting options for the purpose of later re-encryption (by the *VA*1 and the *VA*2), the *VFS* also knows the relation between voting options and the respective ciphertexts. Yet, the *VFS* does not know at which position of the code sheet the cast voting code appears; this information is only known to the *VA*1 and the *VA*2. Consequently, if the *VFS* and either the *VA*1 or the *VA*2 maliciously collaborate, the conspiracy is able to violate vote secrecy of all cast votes.

Voting Device, the NEC, and one Voting Authority. As outlined before, the *VFS* knows the relation between voting options and their respective ciphertexts as well as voters' identities and their cast voting codes. If the *VFS* is not compromised, the adversary might gain the respective knowledge by corrupting voting devices and a threshold of committee members. Throughout the voting phase, voting devices learn the voters' identities and their cast voting codes, while a threshold of the *NEC* is able to decrypt published ciphertexts, thereby constructing the link between voting options and their respective ciphertexts. If this knowledge is associated with either the *VA*1's or the *VA*2's knowledge, the adversary is able to violate vote secrecy of all cast votes.

Vote Integrity. Corrupting the following components allows the adversary to violate vote integrity.

Voting Authorities. Throughout the voting phase, the *VFS* separates the voting code received from the voter and forwards the respective parts to the *VA*1 and the *VA*2. If both authorities the *VA*1 and the *VA*2 agree on selecting the same encryption of a different candidate from the *VSS*, they can undetectably manipulate the voter's cast vote before storing it[25].

Eligibility. Corrupting or controlling the following components allows the adversary to violate eligibility.

Voter, Vote Forwarding Server, and NEC. The first group involves the voter, the *VFS*, and the *NEC*. If the voter forwards her code sheet to the *VFS*, then the *VFS* can cast one voting code from that voter's code sheet. As the *VFS* is not in possession of a valid

[25] Because the *VA*1 and the *VA*2 are not aware of the content they re-encryt, both authorities could merely alter the vote into a random vote, rather than a specific vote.

voter query, the committee members must agree on the eligibility violation. Hence, if one voter collaborates, the adversary can cause the maximum impact on eligibility.

Vote Forwarding Server, Voting Authorities, and NEC. The second group consists of the *VFS*, the *VA*1, the *VA*2, and the committee members. Rather than receiving code sheets from the voters, the *VFS* might receive valid voting codes from the *VA*1 and the *VA*2. In malicious agreement with a threshold subset of the committee members, the group would succeed in violating eligibility for all abstaining voters.

Fairness. Analogously to the Estonian case, only votes not being updated may reliably be revealed by the adversary before the end of the voting phase. Hence, corrupting the following components allows the adversary to violate fairness.

Vote Forwarding Server and one Voting Authority. In accordance to the vote secrecy case, in addition to the *VFS*, the adversary has to corrupt either one of the two voting authorities *VA*1 or *VA*2 in order to calculate intermediate results, thereby violating fairness of all votes that are not updated.

Voting Device, NEC, and Voting Authorities. In analogy to the vote secrecy case, the adversary might corrupt voting devices to determine voting codes have been cast by the voters. If the adversary additionally controls either one of the voting authorities *VA*1 or *VA*2, he is able to relate the cast voting codes to ciphertexts. If the adversary in addition controls a threshold of committee members, these are able to decrypt obtained ciphertexts. The conspiracy is consequently capable of determining votes for those voters that vote over compromised voting devices.

Data Access Protection. According to the specification, the following components learn voter identities and can therefore violate data access protection.

Voting Device. Because of the fact that voting devices are generally known for a number of purposes, voting devices know the voters' identities. A compromised voting device might consequently forward a voter's identity to the adversary.

Distribution Authority. In the setup phase, *DA* provides voters with their code sheets. Therefore, the *DA* knows voters' identities and their postal addresses. In case of corruption, the *DA* might forward these information to the adversary.

Vote Forwarding Server. In analogy to the original Estonian scheme, the voter casts signed ballots, *i.e.* signed voting codes, to the *VFS*. In case of corruption, the *VFS* might abuse its role and forward voter data to the adversary.

Computationally unrestricted and Communication Channel. Analogously to the case of the original Estonian scheme, if the adversary controls the communication channel between the voting device and the *VFS* and additionally is capable of breaking cryptographic primitives, the adversary can determine the identities of voters who cast an Internet vote.

6.5. Comparison of the Qualitative Security Models of the Original and the Extended Scheme

The original Estonian scheme allows an adversary to violate four security requirements by means of compromised voting devices. In contrast, the proposed extension allows an adversary to violate only one security requirement by only controlling voting devices.

In spite of the fact that processes were largely maintained, the extended scheme does not Pareto dominate the original Estonian scheme. Consider the following facts:

If an adversary obtains a voter's code sheet, the adversary can violate vote secrecy by verifying that exactly one of the voting codes have been arrived at a central component of the scheme. In the original Estonian scheme, a voter cannot support the generation of such a proof.

Furthermore, the original scheme requires the collaboration of *eight* components, namely the members of the *NEC* and the *VSS*. Hence, eligibility in the original scheme depends only on the correct behaviour of offline service providers. On the other side, in the case of full corruption of online service providers (namely *RA*) and the total control of voters (namely *VO*), the extended scheme does only need the corruption of *seven* components, namely the *NEC*.

The adequacy of the original and the extended Estonian scheme therefore depends on election-specific quantitative security evaluation.

6.6. Comparison of the Quantitative Security Models of the Original and the Extended Scheme

Election Settings. On the basis of the qualitative security evaluation, the security of the original and the extended Estonian scheme is quantitatively assessed against the four probabilistic adversaries specified in Section 4.2.

In addition to these adversaries, we construct two additional probabilistic adversaries. The first constructed adversary possesses the capability *VO* with certainty (U[1, 1]). The adversary does not possess the capabilities *VD*, *OFSP*, *VI*, and *HCH* (U[0, 0]). Furthermore, the adversary possesses the capability *ONSP* with a uniform probability distribution between 0.1 and 0.2 (U[0.1, 0.2]). Given the qualitative security models of the original and extended Estonian scheme, we expect the original scheme to outperform the extended scheme with regard to vote secrecy against that adversary.

The second constructed adversary possesses the capabilities *ONSP*, *VO* with certainty (U[1, 1]), and does not possess the capabilities *VD*, *VI*, and *HCH* (U[0, 0]). Furthermore, the adversary possesses the capability *OFSP* with a probability of 0.5 (U[0.5, 0.5]). Given the qualitative security models of both schemes, we expect the original scheme to outperform the extended scheme with regard to eligibility against that adversary.

The probabilistic adversaries considered for the quantitative evaluation of the original and the extended Estonian Internet voting scheme are shown in Table 6.3.

Referring to the Estonian parliamentary elections 2015, we consider a number of 899,793 eligible voters and 176,491 expected voters.

Results. The results of the quantitative security evaluation of both schemes are provided in Tables 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9, and are visualized in Figures 6.13, 6.14, 6.15, 6.16, 6.17, 6.18. In addition to the satisfaction degrees, the tables contain the minimum and maximum theoretically possible satisfaction degrees for both schemes: a minimum (respectively maximum) satisfaction degree corresponds to the quantitative evaluation of qualitative security models with the largest (respectively smallest) probability value for all adversarial capabilities.

In spite of the fact that the extended scheme does not Pareto dominate the original scheme, the security of the extended scheme is at least as good as the original scheme against all adversaries defined in Section 4.2 with regard to all security requirements. We first investigate the significance of the achieved improvements with regard to the different election settings. Subsequently, we discuss the evaluation results within the two constructed election settings.

Consider the results of the first election setting as baseline.

It can be noticed that the dominance of the extended scheme becomes more severe with regard to vote integrity, vote secrecy, and fairness if an adversary increases his capabilities with regard to voting device corruption[26]. This observation is explained by the fact that in the original scheme, compromised voting devices are a single point of failure with regard to these requirements. In contrast, the extended scheme prevents an adversary from violating these requirements when only compromising voting devices.

An adversary increasing his capabilities with regard to service provider corruption does impact the original and the extended scheme to a similar extent. In spite of the fact that the scheme extension addresses several single points of failures with regard to service providers, the quantitative result indicates that service provider corruption is not the prevalent threat to security requirements in the considered election setting.

The satisfaction degrees of both the original and the extended schemes remain largely identical if the adversary increases his capabilities to interact with voters. However, it turns out that the difference between the original and extended Estonian scheme significantly decreases with regard to vote secrecy in that setting. This stems from the fact that voters possess code sheets in the extended scheme, which they can forward to the adversary. In collaboration with several other components, this code sheet can serve as a proof about a voter's vote. In contrast, the voter does not receive any receipt that serves as proof about her vote in the original scheme.

In fact, the previous observation is emphasized within our first constructed election

[26] Notice that this severity becomes visible because of the scale differences in the respective Kiviat diagrams.

setting. If the voter forwards any objects/data they have (with a probability of 1) and the probability that an online service provider is compromised is above 0, then, in the original Estonian scheme, vote secrecy can be enforced to a higher degree than in the extended Estonian scheme.

In the case of an absolute corruption of online service providers and absolute voter control in terms of receiving output from voters, the original Estonian scheme enforces eligibility to a slightly higher degree than the extended scheme. This is explained by the fact that in the extended scheme only seven offline service providers might undetectably cast votes for abstaining voters as opposed to eight offline service providers in the case of the original Estonian scheme.

6.7. Summary

The Estonian Internet voting scheme looks back on a long history. Since 2005, Estonians are able to cast their votes for political elections over the Internet. On the foundation of available literature and reviews of the Estonian Internet scheme, we determined qualitative security models of the original Estonian scheme and identified several shortcomings. Among the most prevalent shortcomings, we identified the fact that four out of five security requirements can be violated by an adversary having the capability of compromising voting devices. We consequently addressed this shortcoming of the scheme and proposed an extension. To eliminate the risks caused by compromised voting devices, our proposal implements the concept of code voting, as introduced by Chaum [Cha01]. The proposed extensions did not result result in a Pareto dominating extension of the original Estonian scheme. We therefore conducted a quantitative security evaluation within six election settings. The findings indicate that in the four previously specified election settings, the extended scheme outperforms the original Estonian scheme with regard to four out of five security requirements. Solely, with regard to data access protection, both schemes rely on the trustworthiness of several single components. However, there are specific settings in which the original Estonian outperforms the proposed extension with regard to single security requirements. In conclusion, while the proposed extension is valuable for most election settings, when comparing the original and the extended scheme, the decision finally depends on the concrete target election setting and the relative importance of legally-founded security requirements.

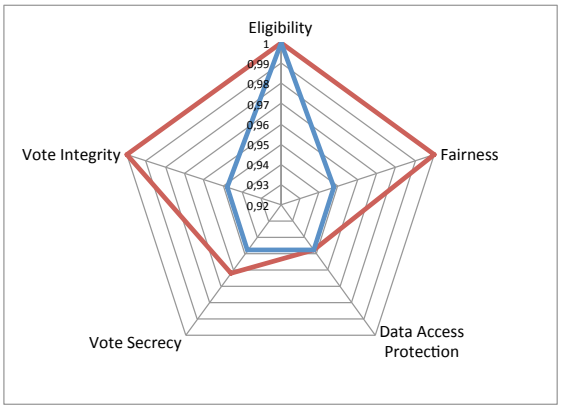

Figure 6.13: Estonia result: Election setting 1.

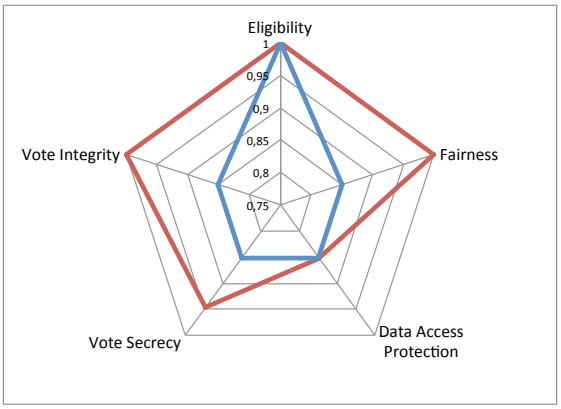

Figure 6.14: Estonia result: Election setting 2.

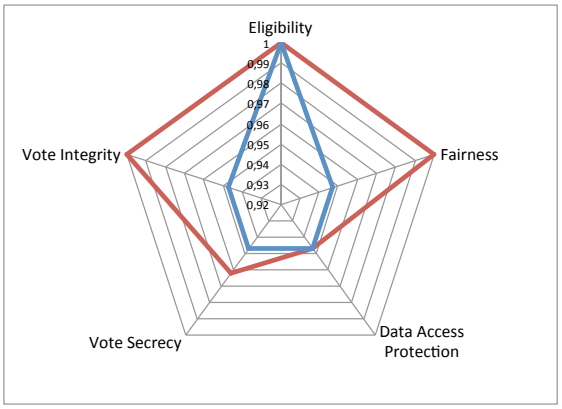

Figure 6.15: Estonia result: Election setting 3.

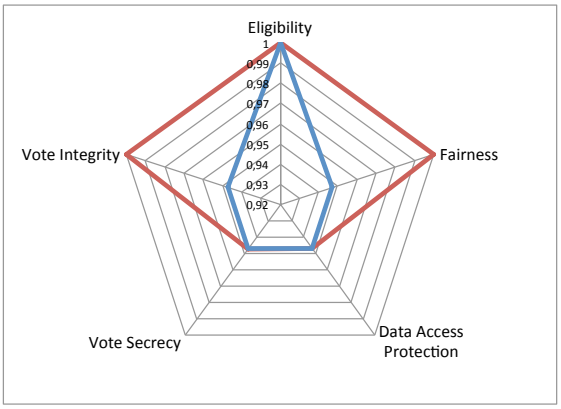

Figure 6.16: Estonia result: Election setting 4.

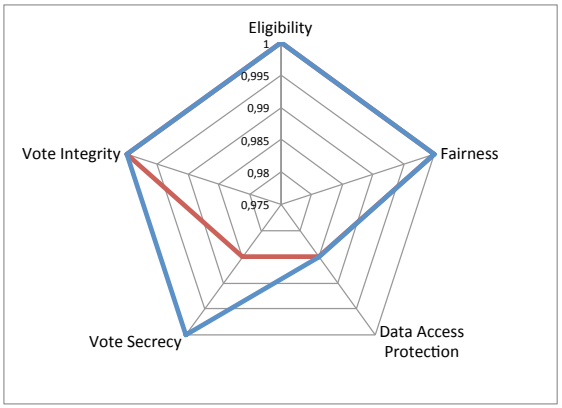

Figure 6.17: Estonia result: Election setting 6.

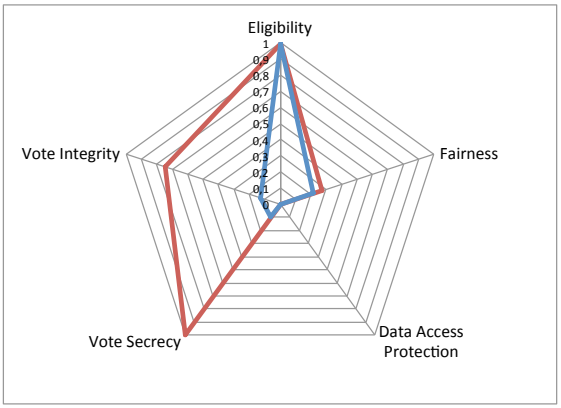

Figure 6.18: Estonia result: Election setting 7.

Requirement	Qualitative Security Models	Impact
Vote Integrity	$(VA1 \wedge VA2)$	$0 \leq l \leq 1$
Eligibility	$\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VFS \wedge (7 \text{ out of } NEC)\Big) \vee$ $(VFS \wedge VA1 \wedge VA2 \wedge (7 \text{ out of } NEC))$	$0 \leq l \leq 1$
Fairness	$(VFS \wedge VA1) \vee (VFS \wedge VA2) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA1 \wedge (4 \text{ out of } NEC)\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA2 \wedge (4 \text{ out of } NEC)\Big)$	$0 \leq l \leq \frac{97}{100}$
Vote Secrecy	$\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge$ $\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VFS\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VA1\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VA2\Big) \vee$ $(VFS \wedge VA1) \vee (VFS \wedge VA2) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA1 \wedge (4 \text{ out of } NEC)\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA2 \wedge (4 \text{ out of } NEC)\Big)$	$0 \leq l \leq \frac{97}{100}$
	$\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VFS\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VA1\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\Big) \wedge VA2\Big) \vee$ $(VFS \wedge VA1) \vee (VFS \wedge VA2) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA1 \wedge (4 \text{ out of } NEC)\Big) \vee$ $\Big(\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \wedge VA2 \wedge (4 \text{ out of } NEC)\Big)$	$\frac{97}{100} < l \leq 1$
Data Access Protection	$\Big(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i)\Big) \vee DA \vee VFS \vee (CR \wedge CCH)$	$1 \leq l \leq n$

Table 6.2: Qualitative security models of the extended Estonian Internet voting scheme.

Election Setting	VD	ONSP	OFSP	VO	VI	HCH
E_1	U[0.01, 0.1]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_2	U[0.1, 0.2]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_3	U[0.01, 0.1]	U[0.01, 0.02]	U[0.001, 0.002]	U[0.01, 0.1]	U[0.01, 0.1]	U[0.01, 0.1]
E_4	U[0.01, 0.1]	U[0.001, 0.002]	U[0.0001, 0.0002]	U[0.1, 0.2]	U[0.1, 0.2]	U[0.1, 0.2]
E_6	U[0, 0]	U[0.01, 0.02]	U[0, 0]	U[1, 1]	U[0, 0]	U[0, 0]
E_7	U[0, 0]	U[1, 1]	U[0.5, 0.5]	U[1, 1]	U[0, 0]	U[0, 0]

Table 6.3: Probabilistic adversaries considered for the quantitative evaluation of the original and extended Estonian scheme.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
Fairness	0.9478738	[0.902240731417, 0.990638176928]	0.999984613	[0.999960900080, 0.999998127716]
DA Protection	0.94787357	[0.902240319619, 0.990638101845]	0.94787357	[0.902240319619, 0.990638101845]
Vote Secrecy	0.94787385	[0.902240822662, 0.990638193589]	0.961904013	[0.902431490663, 0.990696305644]
Vote Integrity	0.9478738	[0.902240731326, 0.990638176920]	0.999999977	[0.999999960000, 0.999999990000]

Table 6.4: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 1.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
Fairness	0.851969229	[0.803114033376, 0.902240777175]	0.999955522	[0.999921253350, 0.999980449099]
DA Protection	0.851968874	[0.803113509064, 0.902240571255]	0.851968874	[0.803113509064, 0.902240571255]
Vote Secrecy	0.851969308	[0.803114149553, 0.902240822868]	0.945994925	[0.902280241805, 0.990640059026]
Vote Integrity	0.851969229	[0.803114033260, 0.902240777152]	0.999999977	[0.999999960000, 0.999999990000]

Table 6.5: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 2.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
Fairness	0.947137063	[0.902239908563, 0.990638026829]	0.999839651	[0.999609359503, 0.999980609731]
DA Protection	0.946794479	[0.902235790545, 0.989010000000]	0.946794479	[0.902235790545, 0.989010000000]
Vote Secrecy	0.947137568	[0.902240795533, 0.990638191114]	0.961984641	[0.902820440441, 0.990713529096]
Vote Integrity	0.947137059	[0.902239899447, 0.990638025996]	0.999997641	[0.999996000000, 0.999999000000]

Table 6.6: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 3.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
Fairness	0.947234111	[0.90224073141, 0.990638176928]	0.999984079	[0.999960900080, 0.999998127716]
DA Protection	0.94723388	[0.902240319619, 0.990638101845]	0.94723388	[0.902240319619, 0.990638101845]
Vote Secrecy	0.947234163	[0.902240822662, 0.990638193589]	0.947249901	[0.902279784527, 0.990639975625]
Vote Integrity	0.947234111	[0.902240731326, 0.990638176920]	0.999999976	[0.999999960000, 0.999999990000]

Table 6.7: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 4.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
Fairness	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]
DA Protection	0.984987414	[0.980000000000, 0.990000000000]	0.984987414	[0.980000000000, 0.990000000000]
Vote Secrecy	1.0000000	[1.000000000000, 1.000000000000]	0.984987414	[0.980000000000, 0.990000000000]
Vote Integrity	1.0000000	[1.000000000000, 1.000000000000]	1.0000000	[1.000000000000, 1.000000000000]

Table 6.8: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 6.

Requirement	Ori. Estonia SD	Ori. Estonia Min/Max SD	Ext. Estonia SD	Ext. Estonia Min/Max SD
Eligibility	0.996093	[0.996093750000, 0.996093750000]	0.9921875	[0.992187500000, 0.992187500000]
Fairness	0.2175598	[0.217559827767, 0.217559827767]	0.2725011	[0.272501147367, 0.272501147367]
DA Protection	0.0000000	[0.000000000000, 0.000000000000]	0.0000000	[0.000000000000, 0.000000000000]
Vote Secrecy	0.0966797	[0.096679687500, 0.096679687500]	1.0000000	[1.000000000000, 1.000000000000]
Vote Integrity	0.1250000	[0.125000000000, 0.125000000000]	0.7500000	[0.750000000000, 0.750000000000]

Table 6.9: Results of the quantitative security evaluation of the original and extended Estonian scheme within election setting 7.

Chapter 7

Conclusion, Limitations, and Future Work

The final chapter summarizes the contributions and limitations of the thesis. Furthermore, we guide future research into several directions.

7.1. Conclusion

This thesis concerned itself with the evaluation of Internet voting schemes and their improvement with regard to legally-founded security requirements. To that end, the thesis sought to answer two research questions.

Research Question 1. *How can the satisfaction of legally-founded security requirements in Internet voting schemes be measured?*

On the basis of previous interdisciplinary work by Bräunlich *et al.* [BGRR13], we refined 13 legal criteria for Internet voting systems into 16 technical requirements at which the implementation of Internet voting systems shall target. On the foundation of the legal instrumental criterion *assurance*, we separated these technical requirements into eight security requirements and eight non-security requirements. The determined technical requirements overcome one shortcoming of Bräunlich *et al.*'s work in reference to our research question: the fact that legal criteria overlap insofar that they capture requirements, measures supporting the satisfaction of requirements, and descriptive refinements. While this distinction is not required in a constructive approach (the designated goal of Bräunlich *et al.*), an overlap in evaluation criteria might result in the fact that certain requirements unintentionally obtain more weight than others which ultimately would lead to questionable evaluation results.

Given the fact that not all legal provisions, and analogously not all deduced technical requirements, can be enforced to their full extent, the German Constitution opens legal latitude to the legislator within which non-ideal voting systems might achieve constitutional compliance. According to the legal latitude, a security evaluation framework for

Internet voting systems, taking the application environment into account, was needed. To construct such a framework for Internet voting schemes (as core building block of Internet voting systems), we determined a set of adversarial capabilities. On the one side, these capabilities serve system analysts to specify election-independent qualitative security models of Internet voting schemes with regard to different security requirements. Furthermore, qualitative security models allow one to capture whether one Internet voting scheme dominates another scheme with regard to security requirements independent of the concrete election setting. On the other side, adversarial capabilities serve election officials to specify election settings in terms of expected adversaries. Because of the non-linearity of qualitative security models and election officials' potential lack of knowledge, election officials might specify these adversaries with uncertainty. The constructed security evaluation framework therefore provides election officials with the possibility to specify adversaries in a probabilistic manner, *i.e.* by assigning probability distributions for the different adversarial capabilities. Upon the specification of qualitative security models of Internet voting schemes and an election setting, the framework evaluates qualitative security models within the election setting by running a large number of Monte-Carlo simulations. The result of this process are satisfaction degrees for Internet voting schemes with regard to legally-founded security requirements, taking the election environment into account.

Research Question 2. *Can established Internet voting schemes be improved with regard to legally-founded security requirements for Internet voting schemes?*

We addressed the second research question by selecting two well-established Internet voting schemes, namely the Polyas Internet voting scheme and the Estonian Internet voting scheme. Both schemes have been used to run a variety of elections and more than two million votes have been cast over both schemes. The qualitative security evaluation of both schemes revealed several shortcomings. The Polyas Internet voting scheme did not maintain vote integrity against compromised voting devices. We addressed this shortcoming by incorporating out of band return codes as means to detect voting devices' misbehaviour throughout the voting process. Upon receipt of the alleged voter intention, the voting service providers return the respective return code(s) to the voting device. Given the fact that a compromised voting device does not learn return codes in advance, the device can only return the return code(s) received from the service providers. The qualitative security models of both schemes demonstrate the Pareto dominance of the extended Polyas scheme over the original scheme. We quantitatively evaluated the security of both schemes in five election settings to evaluate the relevance of the proposed extension in different settings. The results showed that the higher the relative risk of voting device corruption (in relation to other adversarial capabilities), the higher is the relevance of the proposed extension.

The Estonian scheme suffered qualitative shortcomings insofar that four out of five security requirements could be violated by compromised voting devices. We addressed this shortcoming by incorporating out of band voting codes and acknowledgement codes into the original scheme. To prevent single components from violating specific security

requirements due to the incorporation of voting codes, we extended the scheme further towards separation of duties. The extended scheme maintains security against single malicious components (be they voting devices or other components) with regard to four out of five security requirements. In spite of these gains, the proposed scheme does not Pareto dominate the Estonian scheme. Hence, we quantitatively evaluated the security of both schemes in six election settings. In four out of six election settings, the extended scheme performs equally or better than the original Estonian scheme with regard to all security requirements. In two out of six election settings, the original scheme slightly outperforms the extended scheme with regard to one security requirement. These findings indicate that the extended scheme might be the more appropriate scheme from a security perspective in most election settings. However, the selection might also depend on the weighting of different security requirements, which might lead to seldom cases in which the original Estonian Internet voting scheme could be more appropriate in specific election settings.

7.2. Limitations and Future Work

The contributions of this thesis are limited by several assumptions that have been outlined throughout the work. We summarize these limitations in this section and provide thoughts on how these assumptions might be relaxed in the future.

The focus of this thesis are federal elections in the German context. While the legal regulations of this work might indicate a general direction also for other elections, the exact legal regulations might vary from case to case. The investigation of different types of elections would require an interdisciplinary revision of the herein derived technical requirements.

The security evaluation framework targets at the evaluation of Internet voting schemes rather than implemented and running Internet voting systems. From a legal perspective, such a distinction is not made and the elections as a whole have to be conducted in a legally-compliant way. Consequently, when evaluating Internet-based elections, from a technical perspective additional dimensions have to be evaluated, namely the *functions*, *hard-/software*, and *authorities* dimension according to Schryen's reference framework for electronic voting systems [Sch04]. The evaluation of Internet voting systems comes with a further challenge: it has to be determined whether the independence of the scheme layer holds true on the system layer. For instance, the *Vote Forwarding Server* and the *Vote Storage Server* of the Estonian Internet voting scheme are developed and maintained by the same vendor, which practically reduces the security of the scheme's real-world implementation.

The security evaluation framework allows one to evaluate the security of Internet voting schemes, yet, legal regulations prescribe the enforcement of further aspects of Internet voting systems, *e.g.* system accessibility, system usability, and data transparency. Anal-

ogously to the constructed security evaluation framework for Internet voting schemes, similar frameworks for the evaluation with regard to further technical requirements are needed. Of particular importance is the consolidation of scales, *i.e.* differences in the enforcement of security requirements have to be compared against differences in the enforcement of non-security requirements. The evaluation of Internet voting systems with regard to specific requirements, *e.g.* usability and accessibility, is widely built upon established evaluation methods, see for instance the survey on e-voting system usability by Olembo and Volkamer [OV13], and the recommendations by Laskowski *et al.* [LAC+04]. The evaluation of Internet voting systems with regard to other non-security requirements, *e.g.* data transparency, might require further interdisciplinary research.

Given the partially contradicting nature of legal provisions, Internet voting systems –as any other voting method– cannot enforce those provisions to their full extent. The legal latitude provides a means to evaluate the legal compliance of non-ideal voting systems. To evaluate the legal compliance of Internet voting schemes in reference to other voting modes, *e.g.* postal voting, analogous evaluation frameworks for other voting modes are needed.

The security evaluation framework for Internet voting schemes is based on the assumption that voters and the general public verify anything that they can verify (refer to Section 3.1.2). This includes also the fact that voters will use independent verification devices if the scheme foresees their use. In the Polyas scheme, this assumption does also cover that voters check whether they received their election material and whether the seal on the envelope has not been manipulated. Such an assumption does not generally hold true as studies show [KOKV11, KKO+11, OBV13, HW14, AKBW14]. Previous research [NORV14] has shown that people do only take the effort of conducting verification steps, if they are *motivated* and *capable* to conduct these steps. We consider it therefore of fundamental importance to facilitate verification processes and advance scientific research towards voter education. On the other side, we assume that anything that cannot be detected within the scheme, remains undetected. Also this assumptions does not generally hold true. For instance, certain attack patterns might cause suspicion and lead to further investigation. For instance, if exceptionally many votes are cast at the end of the voting phase, it might be an indication for the fact that malicious conspiracies violate eligibility.

The constructed framework allows system analysts and election officials to specify their respective views in a unique manner in terms of eight adversarial capabilities. While these capabilities form a solid starting point for the security evaluation of Internet voting schemes, depending on the application scenario, capabilities might be refined. For instance, operating systems running on voting devices might become part of the evaluation. To that end, system analysts might estimate the relative corruption probabilities for distinct operating systems. Election officials might ultimately only specify the expected number of voters that use different operating systems. Furthermore, the security evaluation framework might be extended by distinguishing central servers from the au-

thorities managing these servers. Election officials would consequently assign corruption probabilities both for servers and authorities.

Currently, the constructed framework considers the presence of adversarial capabilities in a probabilistic manner and the impact caused by different adversaries in a quantitative manner. However, the framework does not consider probabilistic attack strategies, *i.e.* either an adversary is capable of causing certain impact or it is not. However, fine-grained differences in attack strategies become apparent. For instance, certain anonymization techniques allow an adversary to assign certain votes to a subset of all participating voters, see for instance the vulnerabilities outlined by Demirel *et al.* [DJV12]. The framework could be extended by incorporating probabilistic attack strategies.

The constructed framework allows election officials to assess the satisfaction degree of legally-founded security requirements in different Internet voting schemes within the specified election settings. The election official might specify these settings with high uncertainty. In that case, also the computed satisfaction degrees might become highly uncertain. We have addressed this concern by providing minimum and maximum satisfaction degrees for all security requirements within the specified election setting. This direction can be explored further in the future. We currently assume a static adversary model, *i.e.* adversaries have specific capabilities according to specific probability distribution. We do, however, not consider cases in which distributions change over time, *e.g.* adversaries might only gain certain capabilities within a specific time frame. Extending the framework towards dynamic adversaries could potentially lead to higher specification certainty and tighter evaluation results. The underlying Monte-Carlo simulations build a profound basis for uncertainty analysis. The framework could, for instance, provide output distributions, rather than a compressed satisfaction degree. Furthermore, the framework could provide election officials with feedback about the security gains by reducing uncertain capability probabilities or reducing the probabilities of specific capabilities. If the probability that an adversary causes a specific impact with regard to a specific security requirement is linear in all capability probabilities and probabilities are distributed uniformly, then Monte-Carlo simulations can be omitted for the sake of performance. In that case, one can calculate the statistical mean of the uniform distribution and evaluate the probability that an adversary causes a specific impact with regard to a specific security requirement deterministically.

We make the assumption that adversaries gain two different instantiated capabilities with the same probability. This assumption might not always hold true. For instance, developing successful attack strategies – thereby increasing corruption probabilities – against one online/offline service provider or voting device might also influence the corruption probabilities of corrupting other online/offline service providers or compromising other voting devices.

We make the assumption that election officials are capable of providing probabilities for adversaries possessing different assumptions at least with some uncertainty. On an abstract level, there might, however, be further factors influencing those probabilities for

adversarial capabilities, *e.g.* cost-benefit trade offs, funding and expertise.

Within this work, election-specific knowledge has been partially incorporated in the determination of qualitative security models. For instance throughout the evaluation of the Estonian scheme, an estimated number of updated votes and the fact that all citizens using their eIDs for authentication and digital signatures also cast their votes via the Internet, have been used to determine qualitative security models with regard to vote secrecy, fairness, and eligibility. Therefore, in the future, further election-specific knowledge provided by election officials might be used to determine more precise qualitative security models.

The constructed framework supports election officials in identifying the most adequate Internet voting scheme for their election setting from a security perspective. From a practical perspective, decision criteria beyond legally-founded technical requirements might be of relevance, *e.g.* cost considerations, time considerations and trust-enabling measures of Internet voting systems. We therefore recommend to incorporate the security evaluation framework into a larger decision support system for election officials.

The quantitative security evaluation of the Polyas scheme, the Estonian scheme, and their respective improvements are based on the correctness of the determined qualitative security models. To deduce these models, we have reviewed scientific literature, experience reports, and have been in touch with system developers other researchers. Yet, for the future, we recommend a variety of Internet voting schemes to be evaluated by system analysts, *e.g.* JCJ/Civitas [JCJ05, CCM08] and Remotegrity [ZCC+13].

In spite of their qualitative improvements, both the Polyas and the Estonian Internet voting schemes suffer further (and potentially more critical) security shortcomings. For instance, one fundamental problem of the Polyas scheme is the ballot box server's capability to manipulate votes between receiving and storing them. For the future, the herein determined qualitative security models can serve system developers to improve the schemes further towards the enforcements of legally-founded security requirements.

Bibliography

[ABdO76] Ronald N. Allan, Roy Billinton, and Mauricio Figueiredo de Oliveira. An efficient algorithm for deducing the minimal cuts and reliability indices of a general network configuration. *IEEE Transactions on Reliability*, 25(4):226–233, 1976.

[ADMPQ09] Ben Adida, Olivier De Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a university president using open-audit voting: analysis of real-world use of helios. In *2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE)*, pages 10–10. USENIX Association, 2009.

[AKBW14] Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. Usability of voter verifiable, end-to-end voting systems: baseline data for Helios, Prêt à Voter, and scantegrity II. In *2014 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association, 2014.

[Ave85] Terje Aven. Reliability/availability evaluations of coherent systems based on minimal cut sets. *Reliability Engineering*, 13(2):93–104, 1985.

[BC07] Frank Bannister and Regina Connolly. A risk assessment framework for electronic voting. *International Journal of Technology, Policy and Management*, 7(2):190–208, 2007.

[BC14] David Basin and Cas Cremers. Know your enemy: Compromising adversaries in protocol analysis. *ACM Transactions on Information and System Security (TISSEC)*, 17(2):7, 2014.

[Ben06] Josh Benaloh. Simple verifiable elections. In *Electronic Voting Technology Workshop (EVT)*, pages 5–5, 2006.

[BGRR13] Katharina Bräunlich, Rüdiger Grimm, Philipp Richter, and Alexander Roßnagel. *Sichere Internetwahlen: Ein rechtswissenschaftlich-informatisches Modell*. Nomos, 2013.

[BM07] Ahto Buldas and Triinu Mägi. Practical security analysis of e-voting systems. In *Advances in Information and Computer Security*, pages 320–335. Springer, 2007.

[BPM02] Giampaolo Bella, Lawrence C. Paulson, and Fabio Massacci. The verification of an industrial payment protocol: The set purchase phase. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 12–20. ACM, 2002.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[Bun09] Bundesamt für Sicherheit in der Informationstechnik. BSI-CC-PP-0061: Electronic Identity Card (ID_Card PP). Technical report, 2009.

[BWV14] Jurlind Budurushi, Marcel Woide, and Melanie Volkamer. Introducing precautionary behavior by temporal diversion of voter attention from casting to verifying their vote. In *Workshop on Usable Security (USEC)*, 2014.

[CCM08] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. In *2008 Symposium on Security and Privacy (S & P)*, pages 354–368, 2008.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *16th annual international conference on Theory and application of cryptographic techniques*, pages 103–118. Springer-Verlag, 1997.

[Cha01] David Chaum. Sure vote: Technical overview. In *Workshop on Trustworthy Elections (WOTE)*, 2001.

[Che06] Ye Chen. Multiple criteria decision analysis: classification problems and solutions. 2006.

[CL12] Teodor G. Crainic and Gilbert Laporte. *Fleet management and logistics*. Springer Science & Business Media, 2012.

[CMPC13] Marcelo Carlomagno Carlos, Jean Everson Martina, Geraint Price, and Ricardo Felipe Custódio. An updated threat model for security ceremonies. In *2013 Annual ACM Symposium on Applied Computing (SAC)*, pages 1836–1843. ACM, 2013.

[Com07] Technical Guidelines Development Committee. Voluntary voting system guidelines recommendations to the election assistance commission. *Election Assistance Commission*, 2007.

[Com10] Estonian National Electoral Committee. E-voting system. *General Overview*, 2010.

[Cou04] 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, 2004.

[CPP13] Edouard Cuvelier, Olivier Pereira, and Thomas Peters. Election verifiability or ballot privacy: Do we need to choose? In *18th European Symposium on Research in Computer Security (ESORICS)*, pages 481–498. Springer, 2013.

[DJV12] Denise Demirel, Hugo Jonker, and Melanie Volkamer. Random block verification: Improving the Norwegian electoral mix-net. In *5th International Conference on Electronic Voting (EVOTE)*, volume 205 of *LNI - Lecture Notes in Informatics*, pages 65–78. GI, 2012.

[Dre06] Horst Dreier. *Grundgesetz-Kommentar*. Morlok Siebeck Verlag, 2006.

[DS04] Morris R. Driels and Young S. Shin. Determining the number of iterations for monte carlo simulations of weapon effectiveness. Technical report, DTIC Document, 2004.

[DY83] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[EAC09] EAC Advisory Board and Standards Board. Threat trees and matrices and threat instance risk analyzer (TIRA), 2009.

[Ell07] Carl M. Ellison. Ceremony design and analysis. *IACR Cryptology ePrint Archive*, 2007:399, 2007.

[FC05] Eric A. Fischer and Kevin J. Coleman. The direct recording electronic voting machine (DRE) controversy: FAQs and misperceptions. Congressional Research Service, Library of Congress, 2005.

[Fed] Federal Constitutional Court. Decisions of the Federal Constitutional Court (BVerfGE) referred to in this work.

[Fed01] Federal Election Commission. Voting system standards. Technical report, 2001.

[Fel87] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–437. IEEE Computer Society, 1987.

[FTLB01] Olov Forsgren, Ulrich Tucholke, Sébastien Levy, and Stéphan Brunessaux. D4 Volume 3 Report on electronic democracy projects, legal issues of Internet voting and users (i.e. voters and authorities representatives) requirements analysis. CYBERVOTE:WP2:D4V3:2001, EU CyberVote Project. Technical report, 2001.

[Fuq87] Norman Fuqua. *Reliability engineering for electronic design*, volume 34. CRC Press, 1987.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Ges05] Gesellschaft für Informatik. Anforderungen an internetbasierte Vereinswahlen, 2005.

[GJ13] Leonard Gillman and Meyer Jerison. *Rings of continuous functions*. Springer Science & Business Media, 2013.

[GJKR07] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, 2007.

[GKM+06] Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, and Marcel Weinand. Security requirements for non-political internet voting. *Electronic Voting 2006: 2nd International Workshop*, 86:203–212, 2006.

[Gri02] Dimitris A Gritzalis. Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6):539–556, 2002.

[Haz01] Michiel Hazewinkel. *Encyclopedia of Mathematics*. Springer, 2001.

[Hel09] Jörg Helbach. Code Voting mit prüfbaren Code Sheets. In *Informatik 2009: Im Focus das Leben*, volume 154, pages 1856–1862. GI, 2009.

[HJHL11] Axel Hoffmann, Silke Jandt, Holger Hoffmann, and Jan Marco Leimeister. Integration rechtlicher Anforderungen an soziotechnische Systeme in frühe Phasen der Systementwicklung. In *Mobile und ubiquitäre Informationssysteme (MMS)*, volume 185 of *LNI*, pages 72–76. GI, 2011.

[HLV12] Sven Heiberg, Peeter Laud, and Jan Villemson. The Application of I-voting for Estonian Parliamentary Elections of 2011. In *3rd International Conference on E-Voting and Identity (Vote-ID)*, volume 7187 of *Lecture Notes in Computer Science*, pages 208 – 223. Springer, 2012.

[HPR93] Volker Hammer, Ulrich Pordesch, and Alexander Roßnagel. *Betriebliche Telefon- und ISDN-Anlagen rechtsgemäß gestaltet.* Springer, 1993.

[HPW15] Sven Heiberg, Arnis Parsovs, and Jan Willemson. Log analysis of estonian internet voting 2013-2014. In *5th International Conference on E-Voting and Identity (Vote-ID)*, volume 9269 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2015.

[HS07] Jörg Helbach and Jörg Schwenk. Secure Internet Voting with Code Sheets. In *1st International Conference on E-Voting and Identity (Vote-ID)*, volume 4896 of *Lecture Notes in Computer Science*, pages 166–177. Springer, 2007.

[HSS08] Jörg Helbach, Jörg Schwenk, and Sven Schäge. Code Voting with Linkable Group Signatures. In *3rd International Conference on Electronic Voting (EVOTE)*, volume 131, pages 209–208, 2008.

[HW14] Sven Heiberg and Jan Willemson. Verifiable Internet voting in Estonia. In *6th International Conference on Electronic Voting (EVOTE)*, pages 1–8. IEEE Computer Society, 2014.

[IEE05] IEEE. EEE P1583TM/D5.0 Draft Standard for the Evaluation of Voting Equipment, 2005.

[IL00] Sabrina Idecke-Lux. Der Einsatz von multimedialen Dokumenten bei der Genehmigung von neuen Anlagen nach dem Bundesimmissionsschutz-Gesetz, Reihe „Der elektronische Rechtsverkehr“, 2000.

[IW89] Y Iida and H Wakabayashi. An approximation method of terminal reliability of road network using partial minimal path and cut sets. In *Transport Policy, Management & Technology Towards 2001: Selected Proceedings of the Fifth World Conference on Transport Research*, volume 4, 1989.

[JBR+99] Ivar Jacobson, Grady Booch, James Rumbaugh, James Rumbaugh, and Grady Booch. *The unified software development process*, volume 1. Addison-wesley Reading, 1999.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant Electronic Elections. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 61–70. ACM, 2005.

[JFR13] Rui Joaquim, Paulo Ferreira, and Carlos Ribeiro. EVIV: An End-to-end Verifiable Internet Voting System. *Computers & Security*, 32:170–191, 2013.

[JMP09] Hugo Jonker, Sjouke Mauw, and Jun Pang. A formal framework for quantifying voter-controlled privacy. *Journal of Algorithms*, 64(2):89–105, 2009.

[Joa14] Rui Joaquim. How to prove the validity of a complex ballot encryption to the voter and the public. *Journal of Information Security and Applications*, 19(2):130–142, 2014.

[JR07a] Rui Joaquim and Carlos Ribeiro. CodeVoting: Protecting Against Malicious Vote Manipulation at the Voter's PC. In *Frontiers of Electronic Voting*, volume 07311 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[JR07b] Rui Joaquim and Carlos Ribeiro. CodeVoting Protection Against Automatic Vote Manipulation in an Uncontrolled Environment. In *1st International Conference on E-Voting and Identity (Vote-ID)*, volume 4896 of *Lecture Notes in Computer Science*, pages 178–188. Springer, 2007.

[JRF09] Rui Joaquim, Carlos Ribeiro, and Paulo Ferreira. VeryVote: A Voter Verifiable Code Voting System. In *2nd International Conference on E-Voting and Identity (Vote-ID)*, volume 5767 of *Lecture Notes in Computer Science*, pages 106–121. Springer, 2009.

[JRF10] Rui Joaquim, Carlos Ribeiro, and Paulo Ferreira. Improving Remote Voting Security with CodeVoting. volume 6000 of *Lecture Notes in Computer Science*, pages 310–329. Springer, 2010.

[Kal09] Tarmo Kalvet. Management of technology: The case of e-voting in estonia. In *International Conference on Computer Technology and Development (ICCTD)*, volume 2, pages 512–515. IEEE Computer Society, 2009.

[KGK09] Wolfgang Killmann, Alla Gnedina, and Jens Kroder. Health Professional Card (PP-HPC) with SSCD Functionality. Technical Report BSI-PP-0018-V2, 2009. Common Criteria Protection Profile.

[KKO+11] Fatih Karayumak, Michaela Kauer, Maina M. Olembo, Tobias Volk, and Melanie Volkamer. User study of the improved Helios voting system interface. In *1st Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 37–44. IEEE Computer Society, 2011.

[KKY12] Geoffrey Karokola, Stewart Kowalski, and Louise Yngström. Secure e-government services: Protection profile for electronic voting–a case of tanzania. In *Proceedings of the IST-Africa 2012 Conference*, 2012.

[KLP+01] Wolfgang Killmann, Herbert Leitold, Reinhard Posch, Patrick Sallé, and Bruno Baronnet. Secure Signature-Creation Device Type3. Technical report, 2001. Common Criteria Protection Profile.

[KLS96] Jeff Kahn, Nathan Linial, and Alex Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16(4):465–477, 1996.

[KN08] Henry M. Kim and Saggi Nevo. Development and application of a framework for evaluating multi-mode voting risks. *Internet Research*, 18(1):121–135, 2008.

[KOKV11] Fatih Karayumak, Maina M. Olembo, Michaela Kauer, and Melanie Volkamer. Usability analysis of Helios: an open source verifiable remote electronic voting system. In *2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE)*, pages 5–5. USENIX Association, 2011.

[Kru04] Philippe Kruchten. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004.

[KTV11] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Verifiability, privacy, and coercion-resistance: New insights from a case study. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 538–553. IEEE, 2011.

[KTV12] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Clash attacks on the verifiability of e-voting systems. In *2012 Symposium on Security and Privacy (S & P)*, pages 395–409. IEEE Computer Society, 2012.

[LAC+04] Sharon J. Laskowski, Marguerite Autry, John Cugini, William Killam, and James Yen. *Improving the usability and accessibility of voting systems and products*. US Department of Commerce, National Institute of Standards and Technology, 2004.

[Lan10] Lucie Langer. *Privacy and verifiability in electronic voting*. PhD thesis, TU Darmstadt, 2010.

[Lau04] Thomas W. Lauer. The risk of e-voting. *Electronic Journal of e-Government*, 2:177–186, 2004.

[LDEH11] Eric L. Lazarus, David L. Dill, Jeremy Epstein, and Joseph Lorenzo Hall. Applying a reusable election threat model at the county level. In *2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE)*, pages 12–12. USENIX Association, 2011.

[LGTL85] Wen-Shing Lee, Doris L. Grosh, Frank A. Tillman, and Chang H. Lie. Fault tree analysis, methods, and applications: A review. *IEEE Transactions on Reliability*, 34(3):194–203, 1985.

[LKZ14] Huian Li, Abhishek Reddy Kankanala, and Xukai Zou. A taxonomy and comparison of remote voting schemes. In *2014 International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE Computer Society, 2014.

[LLWK10] Kwangwoo Lee, Yunho Lee, Dongho Won, and Seungjoo Kim. Protection profile for secure e-voting systems. In *Information Security, Practice and Experience*, pages 386–397. Springer, 2010.

[LSK12] Jesus Luna, Neeraj Suri, and Ioannis Krontiris. Privacy-by-design based on quantitative threat modeling. In *2012 International Conference on Risk and Security of Internet and Systems (CRiSIS)*, pages 1–8. IEEE Computer Society, 2012.

[Lun10] David Lundin. *Component based electronic voting systems*. Springer, 2010.

[LWW04] Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *Information Security and Privacy*, pages 325–335. Springer, 2004.

[Mat16] Mathematics Stack Exchange. Is the probability of the union of events non-decreasing in the probability of the events? Mathematics Stack Exchange, 2016. URL:http://math.stackexchange.com/q/1636424 (version: 2016-02-02).

[MBT14] Anh Tien Mai, Fabian Bastin, and Michel Toulouse. On optimization algorithms for maximum likelihood estimation. Technical report, CIRRELT Technical Report, 2014.

[McG08] Margaret McGaley. *E-voting: an Immature Technology in a Critical Context*. PhD thesis, National University of Ireland Maynooth, 2008.

[MD13] Theodor Maunz and Günter Dürig. *Grundgesetz: Kommentar*. C.H. Beck, 2013.

[MdSO+15] Taciane Martimiano, Eduardo dos Santos, Maina Olembo, Jean Everson Martina, and Ricardo Alexandre Reinaldo de Moraes. Ceremony analysis meets verifiable voting: Individual verifiability in Helios. In *International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*, pages 105 – 111. IARIA, 2015.

[Mer01] Rebecca T. Mercuri. Electronic vote tabulation checks and balances. 2001.

[MGK02] Lilian Mitrou, Dimitris Gritzalis, and Sokratis Katsikas. Revisiting Legal and Regulatory Requirements for Secure E-Voting. In *International Conference on Information Security (SEC)*, volume 214 of *IFIP Conference Proceedings*, pages 469–480. Kluwer, 2002.

[MM06] Ülle Madise and Tarvi Martens. E-voting in Estonia 2005. The first practice of country-wide binding internet voting in the world. In *2nd International Conference on Electronic Voting (EVOTE)*, volume 86 of *LNI*. GI, 2006.

[MN06] Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In *Advances in Cryptology (CRYPTO)*, volume 4117 of *Lecture Notes in Computer Science*, pages 373–392. Springer, 2006.

[MR10] Niels Menke and Kai Reinhard. Compliance of POLYAS with the Common Criteria Protection Profile-A 2010 Outlook on Certified Remote Electronic Voting. In *4th International Conference on Electronic Voting (EVOTE)*, volume 167 of *LNI*, pages 109–118. GI, 2010.

[MSK+11] Daniel J. Mundform, Jay Schaffer, Myoung-Jin Kim, Dale Shaw, Ampai Thongteeraparp, and Pornsin Supawan. Number of replications required in monte carlo simulation studies: a synthesis of four studies. *Journal of Modern Applied Statistical Methods*, 10(1):4, 2011.

[MU49] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[MV11] Ülle Madise and Priit Vinkel. Constitutionality of remote internet voting: The Estonian perspective. *Juridica International*, 18:4–16, 2011.

[Mye90] Raymond H. Myers. *Classical and modern regression with applications*, volume 2. Duxbury Press Belmont, CA, 1990.

[Myu03] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[NJWS10] Steven Noel, Sushil Jajodia, Lingyu Wang, and Anoop Singhal. Measuring security risk of networks using attack graphs. *International Journal of Next-Generation Computing*, 1(1):135–147, 2010.

[NK06] Saggi Nevo and Henry M. Kim. How to compare and analyse risks of internet voting versus other modes of voting. *EG*, 3(1):105–112, 2006.

[NORV14] Stephan Neumann, Maina M. Olembo, Karen Renaud, and Melanie Volkamer. Helios verification: To alleviate, or to nominate: Is that the question, or

shall we have both? In *International Conference on Electronic Government and the Information Systems Perspective*, volume 8650 of *Lecture Notes in Computer Science*, pages 246–260. Springer, September 2014.

[NV12] Stephan Neumann and Melanie Volkamer. Formal treatment of distributed trust in electronic voting. In *International Conference on Internet Monitoring and Protection (ICIMP)*, pages 30–39. IARIA, 2012.

[NVS+15] Stephan Neumann, Melanie Volkamer, Moritz Strube, Wolfgang Jung, and Achim Brelle. Cast-as-intended-Verifizierbarkeit für das Polyas-Internetwahlsystem. *Datenschutz und Datensicherheit*, 39(11):747–752, 2015.

[OBV13] Maina M. Olembo, Steffen Bartsch, and Melanie Volkamer. Mental models of verifiability in voting. In *4th International Conference on E-Voting and Identity (Vote-ID)*, volume 7985 of *Lecture Notes in Computer Science*, pages 142–155. Springer, 2013.

[Off11] Office for Democratic Institutions and Human Rights. *Estonia Parliamentary Elections, 6 March 2011: OSCE/ODIHR Election Assessment Mission Report*. ODIHR.GAL: Office for Democratic Institutions and Human Rights. ODIHR, 2011.

[OJM11] Samir Ouchani, Yosr Jarraya, and Otmane Aït Mohamed. Model-based systems security quantification. In *2011 Annual International Conference on Privacy, Security and Trust (PST)*, pages 142–149. IEEE Computer Society, 2011.

[OKNV12] Maina M. Olembo, Anna Kahlert, Stephan Neumann, and Melanie Volkamer. Partial Verifiability in POLYAS for the GI Elections. In *5th International Conference on Electronic Voting (EVOTE)*, volume 205 of *LNI - Lecture Notes in Informatics*, pages 95–109. GI, 2012.

[ORBV14] Maina M. Olembo, Karen Renaud, Steffen Bartsch, and Melanie Volkamer. Voter, what message will motivate you to verify your vote. In *Workshop on Usable Security (USEC)*, 2014.

[OSC12] OSCE/ODIHR. Norway: Internet Voting Pilot Project / Local Government Election - 12 September 2011: OSCE/ODIHR Election Expert Team Report., 2012.

[OSV11] Maina M. Olembo, Patrick Schmidt, and Melanie Volkamer. Introducing verifiability in the polyas remote electronic voting system. In *2011 International Conference on Availability, Reliability and Security (ARES)*, pages 127–134. IEEE Computer Society, 2011.

[OV13] Maina M. Olembo and Melanie Volkamer. *E-Voting System Usability: Lessons for Interface Design, User Studies, and Usability Criteria*, chapter 11, pages 172 – 201. IGI Global, February 2013.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology (EUROCRYPT)*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[Pan15] Panda Security. PandaLabs Report Q3 2015 (July - September) 2015. `http://www.pandasecurity.com/mediacenter/src/uploads/2014/07/Pandalabs-2015-T3-EN1.pdf`, 2015.

[Phy04] Physikalisch-Technische Bundesanstalt. Online voting systems for nonparliamentary elections: Catalogue of requirements, 2004.

[PLY10] Harold Pardue, Jeffrey P. Landry, and Alec Yasinsac. A risk assessment model for voting systems using threat trees and monte carlo simulation. In *2009 International Workshop on Requirements Engineering for e-Voting Systems (RE-VOTE)*, pages 55–60. IEEE Computer Society, 2010.

[PLY11] Harold Pardue, Jeffrey P. Landry, and Alec Yasinsac. E-voting risk assessment: A threat tree for direct recording electronic systems. *International Journal of Information Security and Privacy (IJISP)*, 5(3):19–35, 2011.

[PR94] Ulrich Pordesch and Alexander Roßnagel. Elektronisches Signaturverfahren rechtsgemäß gestalten. *Datenschutz und Datensicherheit*, pages 82–91, 1994.

[PYL10] Harold Pardue, Alec Yasinsac, and Jeffrey Landry. Towards internet voting security: A threat tree for risk assessment. In *2010 International Conference on Risk and Security of Internet and Systems (CRiSIS)*, pages 1–7. IEEE Computer Society, 2010.

[Ray08] Samik Raychaudhuri. Introduction to monte carlo simulation. In *Winter Simulation Conference (WSC)*, pages 91–100. IEEE Computer Society, 2008.

[Rep14] Republic of Estonia: Information System Authority. Facts about e-estonia. `https://www.ria.ee/facts-about-e-estonia/`, 2014.

[Ric12] Philipp Richter. *Wahlen im Internet rechtsgemäß gestalten.* Nomos, 2012.

[RJ07] Kai Reinhard and Wolfgang Jung. Compliance of POLYAS with the BSI protection profile - basic requirements for remote electronic voting systems. volume 4896 of *Lecture Notes in Computer Science*, pages 62–75. Springer, 2007.

[RSA78] Ronald L. Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RT13] Peter Y.A. Ryan and Vanessa Teague. Pretty good democracy. In *2013 International Workshop on Security Protocols*, volume 7028 of *Lecture Notes in Computer Science*, pages 111–130. Springer, 2013.

[Sal16] Dietmar A. Salamon. *Measure and Integration.* 2016. To appear in the EMS Textbook series.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in cryptology (CRYPTO)*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1990.

[Sch04] Guido Schryen. Security aspects of internet voting. In *Hawaii International Conference on System Sciences (HICSS)*, pages 9–pp. IEEE Computer Society, 2004.

[Sch09] Wolfgang Schreiber. *Bundeswahlgesetz-Kommentar.* Carl Heymanns, 2009.

[Sec06] Secretariat general de la defense et de la securite nationale. Protection Profile: Machine a voter (PP-CIVIS), 2006.

[SFD+14] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security analysis of the Estonian internet voting system. In *2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 703–715. ACM, 2014.

[SGF02] Gary Stoneburner, Alice Goguen, and Alexis Feringa. Risk management guide for information technology systems. Technical report, National Institute of Standards and Technology Special Publication 800-30, 2002.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[Sha93] Michael I. Shamos. Electronic voting-evaluating the threat. In *Conference on Computers, Freedom and Privacy (CPSR)*, 1993.

[Smi05] Warren D. Smith. Cryptography meets voting. 2005.

[Ste46] Stanley Smith Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

[Str00] Robert S. Strichartz. *The way of analysis.* Jones & Bartlett Learning, 2000.

[SVRH11] Guido Schryen, Melanie Volkamer, Sebastian Ries, and Sheikh Mahbub Habib. A formal approach towards measuring trust in distributed systems. In *2011 Annual ACM Symposium on Applied Computing (SAC)*, pages 1739–1745. ACM, 2011.

[Tre07] Alexander H. Trechsel. *Internet voting in the March 2007 parliamentary elections in Estonia*. PhD thesis, University of Utah, 2007.

[Uni48] United Nations. Universal Declaration of Human Rights, December 1948.

[Vau98] Jussi K. Vaurio. An implicit method for incorporating common-cause failures in system analysis. *IEEE Transactions on Reliability*, 47(2):173–180, 1998.

[Vej13] Martin Vejačka. Evaluation of internet voting systems based on requirements satisfaction. *International Review of Social Sciences and Humanities*, 6(2):41–52, 2013.

[VG09] Melanie Volkamer and Rüdiger Grimm. Determine the Resilience of Evaluated Internet Voting Systems. In *2009 International Workshop on Requirements Engineering for e-Voting Systems (RE-VOTE)*, pages 47–54. IEEE Computer Society, 2009.

[VH04] Melanie Volkamer and Dieter Hutter. From legal principles to an internet voting system. volume 47 of *LNI*, pages 111–120. GI, 2004.

[vMK12] Ingo von Münch and Philip Kunig. *Grundgesetz-Kommentar*. C.H. Beck, 2012.

[Vol09] Melanie Volkamer. *Evaluation of Electronic Voting - Requirements and Evaluation Procedures to Support Responsible Election Authorities*, volume 30 of *LNBIP*. Springer, 2009.

[Vot02] Vote Here. Network Voting System Standards (NVSS). Public Draft 2, 2002.

[VV08] Melanie Volkamer and Roland Vogt. Basic set of security requirements for Online Voting Products. Technical Report BSI-PP-0037, 2008. Common Criteria Protection Profile.

[Web16] WebRoots Democracy. Secure voting: A guide to secure #onlinevoting in elections, 2016.

[ZCC+13] Filip Zagórski, Richard T. Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L. Vora. Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *Applied Cryptography and Network Security (ACNS)*, volume 7954 of *Lecture Notes in Computer Science*, pages 441–457. Springer, 2013.

Appendices

A. KORA Results Derived by Bräunlich et al. [BGRR13]

We outline the results obtained by Bräunlich *et al.* [BGRR13] after execting KORA steps 1-3. Given the fact that KORA steps 1 and 2 are in legal jargon, we provide the results of both steps in the original language German. Given their technical jargon, the technical design goals have been translated to English. We emphasize that legal requirements (KORA step 1), legal criteria (KORA step 2), and technical design goals (KORA step 3) are the result of Bräunlich *et al.* [BGRR13] and are *not* a contribution of this thesis.

A.1. Legal Requirements

Selbstbestimmung. Für ein Internetwahlverfahren lässt sich aus der allgemeinen, unmittelbaren und freien Wahl die Anforderung der Selbstbestimmung ableiten. Jeder Wahlberechtigte muss die Stimmabgabe selbst in Händen halten und sie ohne Hindernisse durchführen können. Dies umfasst zunächst das Recht auf Teilnahme an der Wahl. Für eine obligatorische Internetwahl bedeutet dies, dass jedem Wahlberechtigten ein Zugang zum Verfahren zur Verfügung stehen muss. Das Wahlverfahren muss auch von jedem Wahlberechtigten bedient werden können. Jeder Wähler muss die Möglichkeit haben, durch persönliche Einwirkung im Rahmen seines Wahlrechts seinem Willen gemäßauf das Wahlergebnis Einfluss nehmen zu können. Die höchstpersönliche Stimmabgabe muss auch bei der Internetfernwahl sichergestellt sein, soweit keine Ausnahme für den jeweiligen Wähler besteht. Im Ausnahmefall muss die Möglichkeit gegeben sein, die Stimmabgabe durch eine Vertrauensperson ausüben zu lassen. Der Wähler muss während der Stimmabgabe von unzulässiger Beeinflussung durch das Wahlverfahren frei bleiben. Angesichts tragbarer Minikameras und -computer erscheint es kaum möglich, jegliche mediale Beeinflussung des Wählers während der Stimmabgabe auszuschließen. Entscheidend ist aber, dass durch das Wahlverfahren, das mit dem Ansehen einer staatlichen Einrichtung ausgestattet ist, die Entscheidung des Wählers nicht beeinflusst wird. Dies betrifft auch mediale Beeinflussungen, die in unzulässiger Weise in das Wahlverfahren eingebracht werden. Die zwingende Beeinflussung durch andere Personen wird von der Anforderung Unbestimmbarkeit (A3) adressiert.

Gleichwertigkeit. Für Internetwahlverfahren lässt sich aus der gleichen Wahl die Anforderung der Gleichwertigkeit ableiten. Diese bezieht sich auf das aktive und das passive Wahlrecht. Jeder Wahlberechtigte muss seine Stimme wie jeder andere abgeben können. Jeder Wahlberechtigte darf nur eine wirksame Stimme abgeben, 14 Abs. 4 BWG. Jede gültig abgegebene Stimme muss mit dem gleichen Zählwert in das Ergebnis einfließen. Jeder Wahlbewerber muss in gleicher Weise auf dem elektronischen Stimmzettel dargestellt werden.

Unbestimmbarkeit. Aus der geheimen Wahl lässt sich die Anforderung der Unbestimmbarkeit des Wahlverhaltens ableiten. Kein Wähler darf mit dem Inhalt seiner Stimme in Verbindung gebracht werden. Der Wahlvorgang muss technisch so gestaltet sein, dass es nicht möglich ist, die Wahlentscheidung eines bestimmten Wählers zu erkennen. Eine mündliche Offenbarung durch den Wähler selbst beeinträchtigt die geheime Wahl nicht, soweit der Wahrheitsgehalt einer solchen Aussage nicht überprüft werden kann. Das Wahlergebnis darf nur als Gesamtergebnis nach Ablauf der Wahlzeit bekannt werden. Zwischenstände dürfen nicht bekannt werden, solange noch gewählt werden kann. Der Grundsatz der geheimen Wahl verpflichtet aufgrund seines auch den Wähler verpflichtenden Charakters zu verfahrensrechtlichen und materiellen Vorkehrungen für den Schutz des Wahlgeheimnisses.

Laienkontrolle. Für Internetwahlverfahren lässt sich aus dem Grundsatz der Öffentlichkeit der Wahl die Anforderung der Laienkontrolle ableiten. Die wesentlichen Schritte der Wahl müssen von den Wählern selbst, den Wahlorganen und von allen interessierten Bürgern nachvollzogen und kontrolliert werden können. Dies bedeutet, dass der einzelne Wähler die inhaltlich korrekte Behandlung seiner eigenen Stimme nachvollziehen und kontrollieren können muss. Der verfassungsgemäße Ablauf aller Stimmabgaben muss für alle Bürger nachvollziehbar und kontrollierbar sein. Die Nachvollziehbarkeit darf gerade nicht auf besonderer technischer Sachkenntnis beruhen. Sie muss dem technischen Laien gleichermaßen möglich sein. Es reicht nicht aus, dass die Wähler auf die die generelle Funktionsfähigkeit eines Systems verwiesen sind. Auch eine elektronische Anzeige darüber, dass die abgegebene Stimme korrekt erfasst wurde und in die Auszählung eingegangen ist, genügt nicht.

Datenschutz. Aus der informationellen Selbstbestimmung und dem Fernmeldegeheimnis lässt sich die Anforderung Datenschutz ableiten. Nur solche personenbezogenen Daten dürfen erhoben und verarbeitet werden, die zur Wahldurchführung erforderlich sind. Ihre Verwendung muss auf diesen Zweck beschränkt bleiben. Den Wahlberechtigten muss die übersicht und Kontrolle über ihre erhobenen und verarbeiteten personenbezogenen Daten erhalten bleiben.

A.2. Legal Criteria

Nutzbarkeit (engl. Usability). Die selbstbestimmte Wahl ist nur dann gewährleistet, wenn jeder Wahlberechtigte das Wahlverfahren seinem Willen gemäß bedienen kann. Dem Wähler muss zur Verwirklichung der Selbstbestimmung (A1) die Möglichkeit gegeben sein, seinem Willen gemäß zu wählen, nicht zu wählen oder ungültig zu wählen. Nur dann kann von Gleichwertigkeit (A2) gesprochen werden, wenn alle Wähler durch die Nutzung des Wahlverfahrens das Gleiche erreichen können. Grundsätzlich muss jeder Wähler seine Stimme höchstpersönlich und ohne Hilfe abgeben. Die höchstpersönliche Stimmabgabe muss durch wirksame Maßnahmen auch bei der Stimmabgabe aus dem nichtöffentlichen Bereich sichergestellt werden. Die höchstpersönliche Stimmabgabe muss auch behinderten Wählern möglichst weitreichend durch eine barrierefreie Gestaltung des Wahlverfahrens ermöglicht werden. Es muss jedoch in technisch nicht lösbaren Fällen Wahlberechtigten und ihren Hilfspersonen möglich sein, eine verbleibende Hilfsbedürftigkeit geltend zu machen und die Wahl mittels menschlicher Hilfestellung durchzuführen. Überdies muss die Bedienung der nötigen Anwendungen entweder völlig selbsterklärend sein oder den Wahlberechtigten während des Wahlvorgangs in geeigneter Weise nahegebracht werden. Hierbei muss auch die unterschiedliche Erfahrung der Wahlberechtigten im Umgang mit Rechnern und dem Internet Berücksichtigung finden. Der Schwierigkeitsgrad der Bedienung ist an den Unerfahrensten auszurichten.

Erreichbarkeit (engl. Reachibility). Für eine obligatorische Internetfernwahl muss zunächst sichergestellt sein, dass alle Wahlberechtigten rein physisch die Möglichkeit haben, auf das Verfahren zuzugreifen. Ihnen muss also ein Rechner mit Internetanschluss und den nötigen Anwendungen zur Verfügung stehen. Insofern dies allein durch private Endgeräte nicht gewährleistet ist, müssen öffentliche Geräte zur Verfügung gestellt werden, um die Selbstbestimmung (A1) aller Wahlberechtigten zu ermöglichen. Darüberhinaus müssen die notwendigen Wahldaten wie Zugangsdaten der Wahlberechtigten,
Wählerregister und die Liste der Wahlkandidaten während des Wahlzeitraums in aktuellster Form zur Verfügung stehen, damit jeder selbstbestimmt (A1) und gleichwertig (A2) an der Wahl teilnehmen kann. Jeder Wahlberechtigte muss in die Lage versetzt sein, sich gegenüber dem Wahlverfahren zu identifizieren und sein bestehendes Wahlrecht zu beweisen. Letztlich bezieht sich Erreichbarkeit auf das Wahlverfahren selbst. Es muss während des Wahlzeitraums ohne erhebliche Ausfälle erreichbar sein, um eine selbstbestimmte (A1) Wahlausübung aller Wahlberechtigten zu ermöglichen. Das Verfahren muss daher stabil genug sein, um für den Zeitraum des Wahlvorganges bereitzustehen und die anfallenden Zugriffe der Wähler zu verarbeiten. Es muss die abgegebenen Stimmen so speichern, dass sie auch im Fall von Ausfällen nicht verloren gehen.

Stimmengleichheit (engl. Equality of votes). Die Möglichkeit der wirksamen Stimmabgabe darf nur Wahlberechtigten zur Verfügung stehen, die noch keine verbindliche Stimme

abgegeben haben. Alle Stimmen müssen mit dem gleichen Zählwert in das Ergebnis einfließen. Das Wahlverfahren muss demnach so eingerichtet sein, dass es nur Stimmen von Wahlberechtigten akzeptiert und jede Stimme mit der gleichen Gewichtung und auch nur einmal gezählt wird.

Neutralität (engl. Neutrality). Aus der Selbstbestimmung (A1) und der Gleichwertigkeit (A2) ergibt sich das Kriterium der Neutralität. Eine inhaltliche Beeinflussung der Wähler durch das Wahlverfahren ist auszuschließen. Wahlwerbung oder Aufrufe zu bestimmtem Wahlverhalten dürfen durch das Wahlverfahren oder vermittels des Wahlverfahrens nicht stattfinden. Die Wahlkandidaten sind formal gleich zu behandeln. Niemand ist bei ordnungsgemäßem Verfahrensablauf zu bevorteilen oder zu benachteiligen. Die Darstellung und Menüführung des Wahlverfahrens dürfen nicht die Wahl bestimmter Kandidaten erleichtern oder nahe legen, andere verbergen oder als minderwertig erscheinen lassen.

Unerkennbarkeit (engl. Unknowableness). Aus der Unbestimmbarkeit (A3) und dem Datenschutz (A5) ergibt sich, dass der Inhalt der verbindlich abgegebenen Stimmen von der Stimmabgabe bis zum Ende der Wahlzeit vor Kenntnisnahme geschützt werden muss. Der Inhalt der abgegebenen Stimmen muss zwar zum Auszählen der Stimmen verarbeitet werden. Vorher darf der Inhalt der Stimmen aber niemandem außer dem jeweiligen Wähler selbst zur sicheren Kenntnis gelangen. Zu diesem Zweck bleiben die Stimmzettel bei der Papierpräsenzwahl bis zur Feststellung des Ergebnisses jeglicher Kenntnisnahme entzogen. Ein für die informationstechnische Sphäre vergleichbarer Schutz des Inhalts der Stimme muss auch für ein Internetwahlverfahren eingerichtet werden. Es darf niemandem außer dem einzelnen Wähler in Bezug auf seine eigene Stimme möglich sein, vor Ende des Wahlzeitraums den Inhalt abgegebener Stimmen auszulesen oder auf anderem Weg zur Kenntnis zu nehmen. Bei einer Wahlausübung aus dem privaten, beruflichen oder gesellschaftlichen Bereich muss die individuelle Wahlentscheidung davor geschützt werden, dass Dritte sie ausspähen, sei es durch einfachen Blick auf die Anzeige während der Wahldurchführung, sei es durch lesenden Zugriff auf das Endgerät, sei es durch Mitlesen während der übertragung. Es müssen wirksame Maßnahmen geschaffen werden, um das Wahlgeheimnis auch bei der Wahl aus der privaten Sphäre heraus zu schützen. Unerkennbarkeit schützt gleichzeitig vor unzulässiger Beeinflussung bei der Stimmabgabe, als auch vor der Berechnung und Veröffentlichung von Zwischenergebnissen.

Unverknüpfbarkeit (engl. Unlinkability). Das Kriterium Unverknüpfbarkeit konkretisiert einen weiteren Abschnitt der rechtlichen Anforderung Unbestimmbarkeit (A3) und entspricht dem Datenschutz (A5). Der Inhalt wirksam abgegebener Stimmen darf zu keinem Zeitpunkt, weder während der Stimmabgabe noch im Nachhinein, der bürgerlichen Identität des Wählers zugeordnet werden können. Bis zum Ende der Wahlzeit ist dies durch die Unerkennbarkeit gewährleistet, da der Inhalt der Wahlentscheidung gar nicht von anderen Personen als dem Wähler wahrgenommen werden darf. Unverknüpfbarkeit

setzt mit der Stimmauszählung ein, wenn hierzu die Unerkennbarkeit aufgehoben werden muss. Der nun zu veröffentlichende Inhalt der Wahlentscheidungen darf für niemanden als den Wähler mit den im Wahlverfahren gespeicherten personenbezogenen Daten in Verbindung gebracht werden können. Auch eine Wahrnehmungsmöglichkeit für den einzelnen Wähler selbst darf es nicht erlauben, Dritten die eigene Wahlentscheidung zu beweisen. Auch wenn Unverknüpfbarkeit wirkungsmäßig erst mit der Stimmauszählung einsetzt, ist es wahrscheinlich, dass die Voraussetzungen hierfür bereits weit vorher geschaffen werden müssen. Der Wähler darf den Inhalt seiner abgegebenen Stimme gegenüber Dritten nicht beweisen können. Der Wähler muss auch davor geschützt werden, dass er versehentlich in beweisbarer Form preisgibt, welche Stimme er abgegeben hat. Die Zuordnung von Wahlentscheidung und Wähler wird bei papierbasierten Wahlverfahren grundsätzlich dauerhaft geheimgehalten. Nach Einwurf des Stimmzettels in die Urne ist eine Zuordnung zum Wähler kaum noch möglich. Die Wahlunterlagen müssen gemäß 73 Abs. 2, 89 BWO sicher verwahrt und schließlich gemäß 90 Abs. 3 BWO vernichtet werden. Bezüglich elektronischer Wahlen wird häufig bezweifelt, dass eine solche endgültige Geheimhaltung gewährleistet werden kann, da informationstechnische Daten nur schwer restlos zu löschen sind, die Verbindung zum Wähler nicht so leicht zu trennen ist, wie beim Einwurf in die Papierwahlurne und Verschlüsselungen durch neuere, leistungsfähigere Rechner möglicherweise in kurzer Zeit geknackt werden könnten. Die Verbindung zwischen einem Wähler und seiner Stimme schon bei der Stimmabgabe endgültig zu kappen würde überdies die Möglichkeit des jeweiligen Wählers, eine nachträgliche Kontrolle der ordnungsgemäßen Verarbeitung seiner Stimme durchzuführen (siehe hierzu Individualkontrolle K7), erheblich verkürzen. Eine endgültige Geheimhaltung ist allerdings, auch wenn sie die sicherste Lösung darstellt, nicht notwendig. Die Unverknüpfbarkeit als Konkretisierung der geheimen Wahl dient dem Schutz der freien und gleichen Wahl. Die Schutzfunktion bezüglich der gleichen Wahl erledigt sich mit dem Ende der Wahlzeit. Unzulässiger Zwang aufgrund sicherer Kenntnis der Wahlentscheidung muss aber sowohl bezüglich der aktuellen als auch zukünftiger Wahlen ausgeschlossen werden. Die Unverknüpfbarkeit muss daher so lange gewährleistet werden, wie ein Wähler lebt und an Wahlen teilnehmen kann. Es wäre also möglich, die Verbindung zwischen Wähler und Stimme für die Wahlkontrolle aufrecht zu erhalten. Damit diese Verbindung jedoch von anderen Personen als dem Wähler selbst nicht aufgedeckt werden kann, sollte eine Anonymisierung mit solchen Mitteln durchgeführt werden, die zumindest für die Zeitspanne eines wahlberechtigten Lebens (hier die Zeitspanne zwischen dem vollendeten 18. Lebensjahr und dem Tod) lesenden Angriffen standhalten. Mit Hilfe homomorpher Verschlüsselungsverfahren könnte das Wahlergebnis berechnet werden, ohne die einzelnen Stimmen zu entschlüsseln. In diesem Fall wäre eine fehlende Unverknüpfbarkeit nicht schädlich, da die Unerkennbarkeit weiter gewährleistet bliebe. Unerkennbarkeit und Unverknüpfbarkeit dürfen aber niemals gleichzeitig gebrochen werden. Die Ausführungen über die Dauer der Unverknüpfbarkeit würden sich in diesem Fall auf die Unerkennbarkeit beziehen.

Individualkontrolle (engl. Individual control). Anhand der vom Bundesverfassungsgericht vorgegebenen Adressaten von Öffentlichkeit, dem Einzelnen in Bezug auf die Verarbeitung seiner eigenen Stimme, allen Bürgern in Bezug auf den korrekten Ablauf der Wahl, wird die Anforderung Laienkontrolle (A4) zu den rechtlichen Kriterien Individualkontrolle (K7) und Publikumskontrolle (K8) weiter konkretisiert. Die Individualkontrolle steht dem einzelnen Wähler zu. Er muss kontrollieren können, ob seine Stimme vom Wahlverfahren mit dem von ihm gewollten Inhalt gespeichert und gezählt wird. Zu diesem Zweck kann das Wahlverfahren so eingerichtet sein, dass der einzelne Wähler den Inhalt seiner eigenen Stimme jederzeit, auch nach der verbindlichen Abgabe, einsehen kann. Hierdurch darf nicht die Unverknüpfbarkeit (K6) gebrochen werden.

Publikumskontrolle (engl. Public control). Die Publikumskontrolle richtet sich nicht bloß an die einzelnen Wahlberechtigten, sondern an die gesamte Öffentlichkeit, also auch nicht Wahlberechtigte. Allen Bürgern muss es möglich sein, den verfassungsgemäßen Ablauf jeder Stimmabgabe nachzuvollziehen, also die Einhaltung der Wahlrechtsgrundsätze des Art. 38 Abs. 1. Satz 1 GG. Dabei darf jedoch das Wahlgeheimnis nicht gebrochen, der Inhalt fremder Stimmen also vor dem Ende der Wahl nicht wahrgenommen, nach dem Ende der Wahl nicht mit dem jeweiligen Wähler verknüpft werden (K6).

Datensparsamkeit (engl. Data economy). Aus der Anforderung Datenschutz (A5) und der Anforderung Unbestimmbarkeit (A3) ergibt sich das Kriterium Datensparsamkeit. Menschen können hinsichtlich ihrer personenbezogenen Daten in effektivster Weise geschützt werden, wenn die Daten erst gar nicht erhoben und verarbeitet werden. Der Grundsatz der Datensparsamkeit findet sich zum Beispiel in §3a BDSG. Ein Internetwahlverfahren muss so eingerichtet werden, dass es nur solche personenbezogenen Daten erhebt und verarbeitet, ohne die es nicht funktionieren kann. Die möglichst geringe Verarbeitung personenbezogener Daten entspricht gleichzeitig in hohem Maße der Anforderung Unbestimmbarkeit (A3), denn je mehr personenbezogene Daten verarbeitet werden, desto leichter wird es, das Wahlgeheimnis zu brechen.

Datentransparenz (engl. Data transparency). Als Konkretisierung der Anforderungen Datenschutz (A5) und Unbestimmbarkeit (A3) ergibt sich das Kriterium der Datensparsamkeit. Der Grundsatz der Datensparsamkeit findet sich zum Beispiel in §3a BDSG. Ein Internetwahlverfahren ist so einzurichten, dass es nur solche personenbezogenen Daten erhebt und verarbeitet, ohne die es nicht funktionieren kann. Der bereits im Volkszählungsurteil angelegte Grundsatz der Datensparsamkeit zielt auf eine Vorsorge zur Minimierung der Risiken für die informationelle Selbstbestimmung. Da einmal erhobene personenbezogene Daten unter den Umständen der modernen Datenverarbeitung häufig kaum noch zu kontrollieren sind, wird Datensparsamkeit teilweise sogar zur einzigen Möglichkeit, um informationelle Selbstbestimmung überhaupt noch ausüben zu können. Das Prinzip der Datensparsamkeit dient nicht wie das Erforderlichkeitsprinzip

der Begrenzung von Grundrechtseingriffen im Einzelfall, sondern leistet Vorsorge, indem es eine technisch-organisatorische Gestaltung von Datenverarbeitungsanlagen verlangt, die möglichst keine oder möglichst wenig personenbezogene Datenverarbeitung ermöglicht. Indem möglichst wenige personenbezogene Daten über die Teilnahme an der Internetwahl verarbeitet werden, wird auch der Unbestimmbarkeit (A3), die sich auf das Wahlgeheimnis bezieht, in hohem Maße Rechnung getragen. Je weniger personenbezogene Daten im Zusammenhang mit der Wahl erhoben werden, desto schwieriger wird es, eine abgegebene Stimme einem Wähler zuzuordnen. Die Datensparsamkeit weist also einen erheblichen Bezug zur Unverknüpfbarkeit (K6) auf und ist daher ein sehr wichtiges Kriterium für den Schutz des Wahlgeheimnisses.

Zweckbindung (engl. Appropriation). Zur Verwirklichung des Datenschutzes (A5) ist die Zweckbindung notwendig. Die personenbezogenen Daten dürfen ohne die Einwilligung des Betroffenen vom Wahlverfahren nicht zu anderen Zwecken als zur Wahldurchführung gespeichert oder sonst verarbeitet werden. Nach Erreichung des spezifischen Zwecks innerhalb des Wahlverfahrens sind die personenbezogenen Daten umgehend zu löschen. Das Kriterium Zweckbindung ist verschränkt mit dem Kriterium der Datentransparenz. Den Wahlberechtigten muss ersichtlich sein, zu welchen Zwecken die über sie erhobenen Daten genutzt werden können. Dieses Prinzip des Datenschutzrechts findet sich zum Beispiel in §§4 Abs. 3 Nr. 2, 14 Abs. 1 Satz 1 BDSG und §12 Abs. 2 TMG. Es enthält das Verbot der Datensammlung auf Vorrat54 und der Bildung von Persönlichkeitsprofilen. Die Zweckbestimmung muss konkret und eng gefasst sein, um sicherzustellen, dass sie nicht durch einen zu weit gefassten Zweck umgangen werden kann. Personenbezogene Daten dürfen vom Wahlverfahren nur verarbeitet werden, soweit dies zur Durchführung der Wahl erforderlich ist. Die Verarbeitung darf ohne Einwilligung des jeweiligen Wählers nicht im Nachhinein für andere Zwecke ermöglicht werden.

Datenbeherrschbarkeit (engl. Data controllability). Zur Verwirklichung des Datenschutzes (A5) muss jeder Wähler Einfluss auf die im Rahmen der Internetwahl über ihn gespeicherten personenbezogenen Daten nehmen können. Dies ist durch Löschungs-, Berichtigungs- und Sperrungsrechte zu verwirklichen, wie etwa §20 BDSG. Das Internetwahlverfahren muss dem Wähler eine Möglichkeit bieten, diese Rechte effizient auszuüben.

Sicherung (engl. Assurance). Die Sicherung ist ein Instrumentalkriterium. Sie gewährleistet die Verwirklichung der übrigen Kriterien im Angesicht von Angriffen und Funktionsfehlern. Das Internetwahlverfahren muss nicht lediglich im Normalbetrieb die übrigen zwölf Kriterien erfüllen. Es muss auch so gesichert werden, dass es nicht durch unbefugte Einwirkungen oder Fehler ihnen zuwiderlaufend funktioniert. Zum Schutz der Wahlrechtsgrundsätze sind technische und organisatorische Sicherungsmaßnahmen gegen unbefugte Einwirkungen und Funktionsfehler einzurichten. Rechtliche Sicherungsmaßnahmen, wie

das Wahlprüfungsverfahren, sind möglichst zu unterstützen. Hierzu ist ein das Internetwahlverfahren in seiner Gesamtheit sicherndes Konzept notwendig. Aufgrund der dargestellten erheblichen Risiken, die vom Einsatz der Internetwahltechnik für die Erfüllung des Wahlrechts ausgehen, kommt der systematischen Sicherung der Internetwahl besondere Bedeutung zu. Das Wahlverfahren muss in unangefochtenem und in angefochtenem Zustand den Wahlrechtsgrundsätzen entsprechen. Aufgrund der dargestellten Risiken, die teilweise gleich mehrere Wahlrechtsgrundsätze bedrohen, teilweise erst im Zusammenspiel ihre besondere Gefährlichkeit entwickeln, reicht es aber nicht aus, jeden Wahlrechtsgrundsatz, jede rechtliche Anforderung und jedes Kriterium einzeln zu sichern. Vielmehr bedarf es einer Sicherung, die das ganze Verfahren, alle Risiken sowie alle Unterschiede und überschneidungen im Sicherungsbedarf insgesamt in den Blick nimmt. Bezüglich der personenbezogenen Daten wurde dieser Schutzbedarf bereits in Rechtsnormen ausgestaltet. §9 Satz 1 BDSG schreibt vor, dass im Verhältnis stehende technische und organisatorische Maßnahmen zum Schutz personenbezogener Daten von der verantwortlichen Stelle getroffen werden müssen. In der Anlage zu §9 BDSG sind spezifische Maßnahmen zum Schutz der Daten vorgegeben. Diese können auf Ebene der technischen Gestaltungsziele oder Gestaltungsvorschläge eingebracht werden. Für das Wahlprüfungsverfahren wird überdies eine nachvollziehbare und beweissichere Aufbewahrung der abgegebenen Stimmen und der übrigen erheblichen Wahldaten, sowie eine Dokumentation des Wahlablaufs benötigt. Für die Bundestagswahl als Papierwahl findet sich dieser Anteil des Sicherungskriteriums bisher in den §§72 und 73 BWO konkretisiert.

A.3. Technical Design Goals

Bräunlich *et al.* [BGRR13] derived the following technical design goals for Internet voting systems as result of their interdisciplinary research.

TDG 1: Unauthorized parties must not have the possibility to view voter data.

TDG 2: Unauthorized parties must not have the possibility to manipulate voter data.

TDG 3: Only data required shall be stored.

TDG 4: Any voter must have the possibility to view and influence both extent and purpose of stored her personal data.

TDG 5: The ballot must be neutral.

TDG 6: Unauthorized parties must not have the possibility to change the ballot data.

TDG 7: The election committee must start the election at the predetermined time.

TDG 8: After a system failure, it must be possible to resume the election.

TDG 9: The election committee must stop the election at the predetermined time.

TDG 10: The calculation of intermediate results must not be possible.

TDG 11: The calculation of the election result must be started after the official voting phase by members of the election committee.

TDG 12: Only eligible voters may access successfully the Internet voVting system.

TDG 13: Eligible voters may cast only one binding vote.

TDG 14: The essential steps of the vote casting process must be understandable to any voter.

TDG 15: Any voter must be able to conduct the vote casting process.

TDG 16: All voters must obtain the same result with equal usage.

TDG 17: Eligible voters must have the possibility to cast votes at any time of the voting phase.

TDG 18: The vote may only be cast and stored after a confirmation by the voter.

TDG 19: It must be ensured that the vote is correctly transmitted.

TDG 20: Any voter must receive a message regarding the (non-)success of her voting process.

TDG 21: A voting note must only be taken after a binding vote has been cast.

TDG 22: Third parties must not be capable of linking a vote to the voter who cast the respective vote.

TDG 23: The voter must not be capable of proving her vote to any third party.

TDG 24: It must not be possible to manipulate the stored binding votes.

TDG 25: The system must compute the correct result.

TDG 26: It must not be possible to manipulate the election result.

TDG 27: Any voter must be able to verify that her vote has been included in the election result.

TDG 28: The public must be able to verify that the election result has been derived correctly.

TDG 29: The election must be protocolled.

TDG 30: The election data must be archived in a traceable and evidence-proven manner.

B. Implementation and Graphical User Interface of the Security Evaluation Framework

The security evaluation framework has been implemented by a student assistant at the Technische Universität Darmstadt. The framework has been implemented as JavaFX FXML application version 8 using the build manager Apache Maven. The graphical user interface has been implemented with the Java Layout Manager MigLayout and using Cascading Style Sheets. The backend of the application contains a *MySQL* database. The *EclipseLink ORM* framework is used to map objects in the database.

The graphical user interface of the implemented security evaluation framework is shown in Figures 1 and 2. The interface shown in Figure 1 allows an election official to specify her election setting: Therefore, the election official provides distributions with which the adversary has different capabilities. In the current implementation, the election official specifies uniform distributions. Additionally, the election officials indicates the number of expected voters and the number of eligible voters. Optionally, the election official indicates the number of Monte-Carlo iterations to be run and the target confidence level for the resulting satisfaction degree. If the election official does not specify these values, $10,000$ Monte-Carlo iterations are run with a confidence value of $\approx 95,5\%$.

A typical security evaluation result for two Internet voting schemes is shown in Figure 2. The results are visualized within chart and additionally provided in tabular format. In addition to the satisfaction degrees, the election official obtains further statistical information regarding the information including the confidence value for the computed satisfaction degree and the minimal and maximal possible satisfaction degrees within the specified election setting.

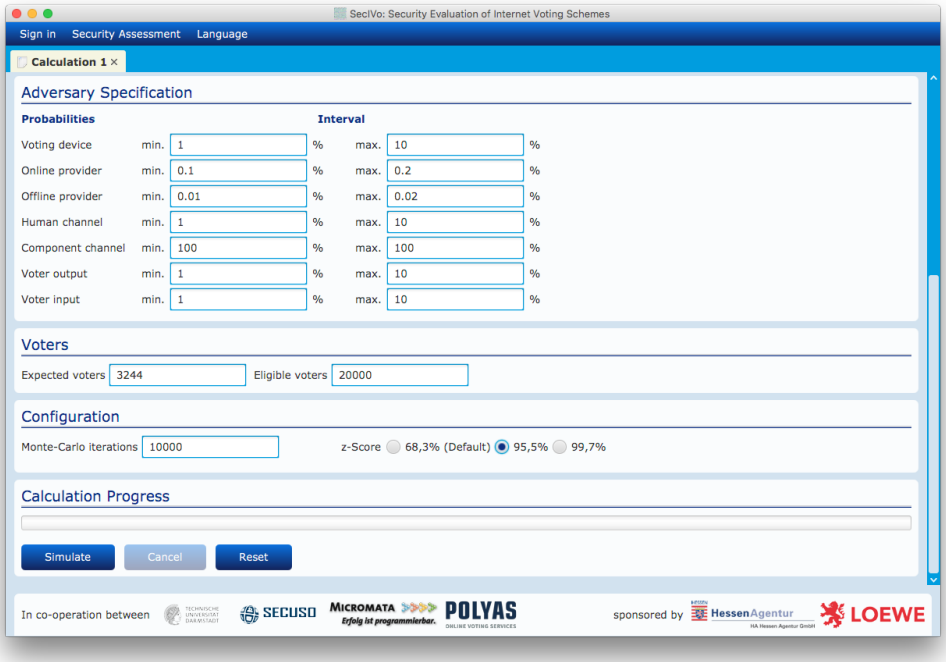

Figure 1: Interface for the election setting specification in the security evaluation framework.

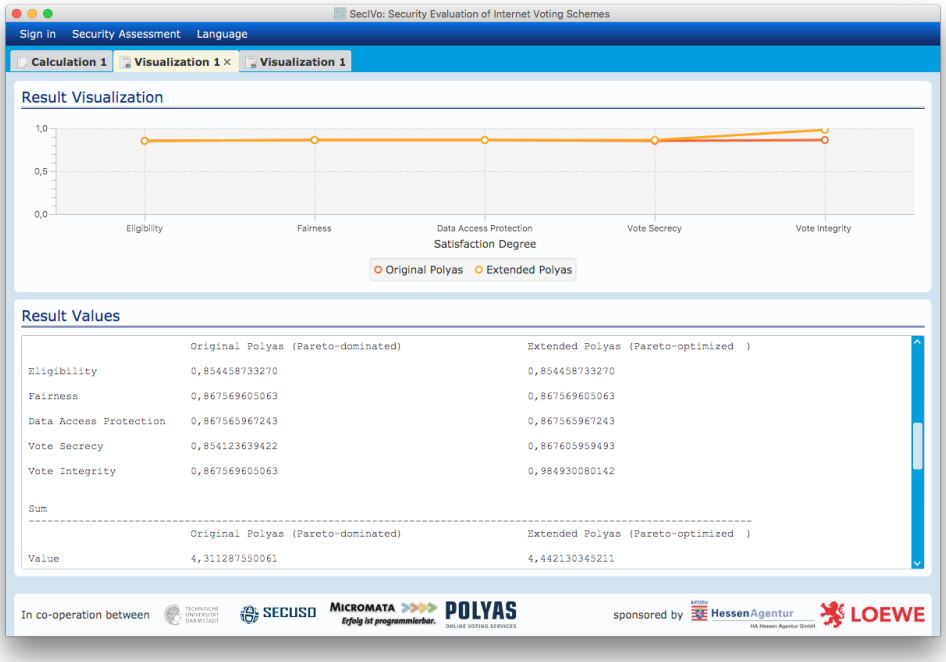

Figure 2: Result interface of the security evaluation framework.

Wissenschaftlicher Werdegang

Oktober 2011 – März 2016

Promotion im Fachbereich Informatik der Technischen Universität Darmstadt unter wissenschaftlicher Leitung von Prof. Dr. Melanie Volkamer

Oktober 2004 – Oktober 2011

Studium der Informatik an der Universität des Saarlandes

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit - abgesehen von den in ihr ausdrücklich genannten Hilfen - selbständig verfasst habe.

Darmstadt, Februar 2016