Department of Informatics
University of Fribourg (Switzerland)

Electronic Voting over the Internet
The Boon and Bane of Modern E-Society

Thesis

presented to the Faculty of Science of the University of Fribourg (Switzerland)
in consideration for the award of the academic grade of
Doctor scientiarum informaticarum

by

Reto E. Koenig

from

Bern, Switzerland

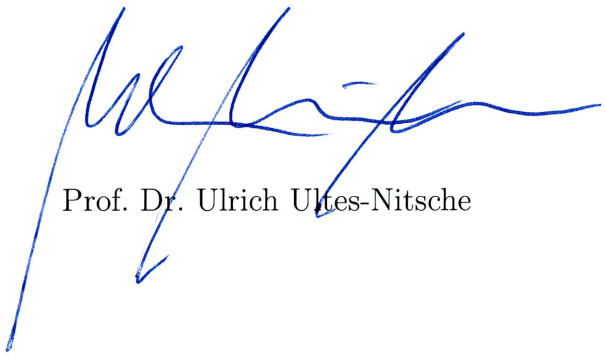
Thesis No. 1798
UniPrint, Fribourg
2013

Accepted by the Faculty of Science of the University of Fribourg (Switzerland) upon the recommendation of:

- Prof. Dr. Ulrich Ultes-Nitsche, University of Fribourg (thesis supervisor),
- Prof. Dr. Rolf Haenni, Bern University of Applied Sciences,
- Prof. Dr. Peter Y. A. Ryan, University of Luxembourg.

Fribourg, June 13, 2013

Thesis supervisor:



Prof. Dr. Ulrich Ultes-Nitsche

Dean:

Prof. Dr. Fritz Müller

to Kati*

*She, who treated me well even in my darkest hours! She who took care for the whole family while I was on my crypto-ego-tripp! Thank you!

Acknowledgment

I want to thank all the people without whom this thesis would not have been possible:

First of all I am deeply grateful to my supervisors Ulrich Ultes-Nitsche, Rolf Haenni and Eric Dubuis. Ulrich-Ultes Nitsche guided me into the academic field whereas Rolf Haenni introduced the world of cryptography to me. It was Eric Dubuis who accepted me as a member within the e-voting group, which stated the starting point of my thesis. This complementary team offered me the highest level of support, always directed towards my academic career, so I could focus on the thesis during the past years.

I am particularly grateful to my fellow colleague Stephan Fischli for drilling me in the subject of precision and lecturing. I very much appreciated the intensive and fruitful discussions and the deep insights I was able to gain.

I further want to thank my fellow PhD-students Oliver Spycher and Michael Schlaepfer. We were given the unique opportunity to challenge each other in a very constructive way for the past four years and we had very insightful discussions which allowed me to broaden my horizon in IT-security.

I very much appreciated the warm and friendly welcome within the e-voting community where I felt accepted from the very beginning. A very special thank goes to Peter Y. A. Ryan who accepted me exceptionally for the Dagstuhl workshop in 2011. This workshop stated one of the corner stones within my thesis.

A special thank goes to Stephan Krenn for proof reading this thesis and for always lending me a helping hand concerning mathematics and cryptography.

I want to thank the following members of the telecommunication, network and security group at the University of Fribourg, and the RISIS-Institute at Bern University of Applied Sciences: Severin Hauser, Joël Allred, Carolin Latze and Ronny Standke. Simon Klaus, Danjel Brei, Andrea Pellegrine, Philémon von Bergen, Daniel Weibel, Pascal Gremaud, Louis Bernath, Jan Thomas Liechti and Christian Lutz who I was pleased to supervise with works spawning from my thesis during their Bachelor's and Master's theses, respectively.

A warm thank you to Heidrun Ulties-Nitsche who urged me gently to write a true German abstract and to dedicate the thesis to the one who really earned it.

The financial support of the Hasler Foundations is gratefully acknowledged.

Abstract

This thesis presents the findings and contributions made during my time as a PhD student. I have been given the opportunity to participate in the struggle of the international e-voting community, to realize a user-friendly, robust and cryptographically secure end-to-end (E2E) verifiable coercion-resistant e-voting system over the Internet, providing true longterm privacy. In short: "The tantalizing quest for the Holy Grail of modern e-society".

Following an introduction to e-voting and the presentation of a security relevant finding in e-voting schemes over the Internet using threshold blind signatures, this thesis demonstrates solutions on multiple levels in order to render coercion-resistant E2E-verifiable e-voting over the Internet finally practical. In particular, one contribution solves the seemingly inherent lack of board flooding resistance at the protocol level, posing a true menace for democracy. This renders the protocol efficient in theory and practice. However, the proposed solution aggravates the problem on the voter side, where the voter is requested to securely remember and discriminate multiple high entropy credentials in such a way that the adversary is oblivious to them. A follow-up contribution gives a solution for this exact problem by introducing a key-management primitive, enabling the voter to manage high entropy credentials within a single ciphertext that is fully deniable in the presence of coercion. At this point, the thesis introduces a relaxed voter-model, allowing to model unintended and unrecognized errors made by the voter while processing high entropy credentials. It is then shown, that all but one coercion-resistant scheme fail to provide the property of individual verifiability "*counted as intended*" when applying the relaxed voter-model. Furthermore this thesis contributes towards solving the secure platform problem in remote e-voting. With the use of an offline voting device presenting special abilities but restricted computing power and minimal trust assumptions on the voter's side, it demonstrates how to preserve voter anonymity, and privacy during voting and verifying. It is demonstrated, that this solution gains maximum voter-usability if the vote is composed on an explicitly untrusted but computationally powerful device, such as the voter's PC. Combining the contributions finally enables a scetch of productive systems usable under the harsh conditions of reality. However, the problem of everlasting privacy remains.

Zusammenfassung

Dies Arbeit präsentiert die Ergebnisse, zu denen ich während meiner Zeit als Doktorand auf dem Gebiet des elektronischen Wählens (E-Voting) beigetragen habe. Sie repräsentieren die aktive Beteiligung am Versuch der Erschaffung eines benutzerfreundlichen, robusten und kryptographisch sicheren Ende-zu-Ende (E2E)-verifizierbaren E-Votings mit Erpressungsresistenz, zur Durchführung demokratischer Abstimmungen und Wahlen über das Internet.

Nach der Einführung ins E2E-verifizierbare E-Voting, wird ein Sicherheitsaspekt im Zusammenhang beschrieben, der bei blinden Signaturen mit Schwellwert auftritt. Dann befasst sich die Arbeit mit der Praxistauglichkeit von erpressungsresistenten, E2E-verifizierbaren E-Voting Protokollen. Ein Beitrag löst das auf Protokolleblene scheinbar inherente Fehlen des Flutungsschutzes für das elektronische Anschlagbrett; ein Umstand, der bisher eine Gefahr für die Demokratie darstellte. Dies garantiert praxistaugliche Effizienz. Andererseits verschlechtert sich dabei die Benutzerfreundlichkeit auf Seiten der Wählerschaft. Diese muss nun mehrere hochentropische Berechtigungsschlüssel sicher verwalten- und unterscheiden können, uneinsehbar für mögliche Angreifer. Dafür wird eine kryptograpfische Primitive zur abstreitbaren Verwaltung beliebig vieler Berechtigungsschlüssel eingeführt. Dabei wird ein relaxiertes Wählerschaftsmodell eingeführt, welches ungewollte oder unbemerkte Fehler abbildet, die den einzelnen Mitgliedern der Wählerschaft unterlaufen können. Dieses lässt die meisten bisher beschriebenen erpressungsresistenten E2E E-Voting Schemen brechen, da die Eigenschaft der individuellen Verifizierbarkeit "*gezählt wie vorgesehen*" verloren geht. Der Letzte Beitrag bietet eine Lösung des "sichern Plattform Problems", welche während der Stimmabgabe über das Internet zum Zuge kommt. Es wird ein offline-Wahlgerät beschrieben, das spezialisierte Fähigkeiten aufweist, jedoch nur eingeschränkte Verarbeitungsmöglichkeiten hat. Dies garantiert den Schutz der Privatsphäre während der Abstimmung unter geringsten Vertrauensannahmen. Zusammen mit dem nicht-vertrauenswürdigen Computer wird eine sehr hohe Benutzerfreundlichket erreicht. Die Kombination aller Beiträge erlaubt die Skizzierung eines produktiven Systems, welches unter den rauen Alltagsbedingungen eingesetzt werden kann. Dennoch, das Problem der garantierten Wahrung der Privatsphäre bleibt bestehen.

Contents

Chapter 1

Introduction

"It is as if we were extracting bullets from a loaded revolver"
This was the internal concluding statement at the 2011 e-voting workshop in Dagstuhl (Germany), which I had the honour to attend. Paul Gibson started this metaphor in order to describe the work of cryptographers when it comes to the subject of e-voting. He compared e-voting to the potential of a loaded revolver[1]: Once released, someone will eventually grab it and explore its features. Hence, the only possibility left for cryptographers and computer scientists to prevent the final disaster is to extract as many bullets as possible from it *before* e-society gets a hold of it. Although the possibility of a sudden bang is reduced with every bullet extracted, the tragic end of the game is predetermined as long as a final bullet remains. So, even if the extraction process is tedious and may still be far from finished, this weapon of power should not be released until the last bullet is safely stowed away. If for any reason, some bullets just cannot be extracted, e-voting should not be released at all, otherwise you might find yourself in the line of fire. Directed at a whole nation, this ultimately raises the question of national security.

In this spirit, this dissertation gives an insight into the cumbersome process of trying to unload the heavy gun called e-voting. It is about the transformation of the seemingly simple act of democratic voting into a cryptographically secure and universal end-to-end verifiable e-voting system that resists the pressure of power, remains trustworthy in moments of greatest surprise and always supports democracy. The thesis is not about cryptographers' self-adulation, but their ultimate goal of handing over an end-user aware solution not bearing the inherent threat of a destructive bang against the democratic world, when someone eventually pulls the trigger.

[1] Does Russian roulette ring a bell?

1.1. Variants of E-Voting

The term e-voting is used as an umbrella term covering different technologies and ranges of applications. The two most prominent e-voting technologies are *supervised voting* and *remote voting* [96]. They cover two extreme ranges of applications. Supervised voting comprises electronic voting machines acting as trusted voting devices that are located within an isolated voting booth in a dedicated polling station. Remote voting on the other hand implies voting over the Internet, enabling the voter to cast the vote from anywhere (not guaranteeing isolation) using ordinary computers—highly untrusted equipment.

Both variants come in many shades: Blackbox e-voting systems on the one side cover systems where no one (but the system maintainer) has knowledge about the machinery and program-code in action. Whitebox e-voting systems on the other hand, give full insight to the machinery and program-code that will be used. Even though whitebox systems may have a significant impact in trust, they do not necessarily improve the confidence in the final tally. This comes as a conclusion to a source code analysis we were invited to do for an existing e-voting system used for democratic voting in Switzerland. So, it does not really matter what shade of gray a system in action represents, neither of which can give true confidence in the final tally.

"Verify the election, not the system!" Josh Benaloh once said, meaning that in order to guarantee the correctness of a voting system (no matter which color), the outcome of the election itself must be verifiable. In its purest form, the outcome allows to mathematically verify the complete process from the moment of vote-casting up to the final tally, without tampering privacy or the secrecy of the vote. This form of end-to-end verifiability is therefore labeled as *E2E-verifiable e-voting* and allows to give a mathematically based degree of confidence to the final tally.

This thesis mainly concentrates on E2E-verifiable remote voting. Any further use of the term *e-voting* will refer to only this technology and to this range of application.

Furthermore, in this thesis the term *voting*[2], will be used as an umbrella term also comprising *election*[3]. The term *e-voting scheme* refers to the mathematical models and descriptions of an e-voting protocol, whereas the term *e-voting system* refers to a true implementation runnable on real computers. The main focus of this dissertation is on e-voting schemes.

[2]deciding about objects
[3]deciding about subjects

1.2. Lift-Off

Even though it seems as if the idea of remote e-voting using an omnipresent computer is a phenomena of the late 1990's in the dawning light of the Internet, this assumption is wrong. The very first concrete description of a remote e-voting scenario was provided in 1955 by Isaac Asimov, a very famous science fiction author. The short story entitled *Franchise* [5] describes the voting process within an e-society where all decisions are made by algorithms present in a gigantic supercomputer called *Multivac*. The system requires the "opinion" of a single citizen called *Voter of the Year*, chosen by Multivac itself in order to bring in randomization. The story gives critical answers to the question of why the described society switched to e-voting: the chronic impatience of the society to wait for the final tally, getting rid of the time-consuming queueing-up in front of polling-stations and the possibility (from an engineer's perspective) to blame a bad decision made by Multivac to the 'Voter of the Year' who 'initialized' it. Long story short: This system was mainly motivated by convenience and cost-reduction.

Listening to the Zeitgeist, it seems as if Asimov just hit the bull's-eye. However, there is yet another motivation for e-voting to be considered, which in its most extreme extent sounds rather like science fictional today: It covers voting from abroad, allowing citizens to cast their vote no matter where they live physically. Please note that this requirement is not restricted to the planet itself. In 1997 the Texas (USA) bill granted the right to vote to astronauts on long-term missions. David Wolf was the first subject executing this right by casting his ballot from the Russian Space Station Mir and more recently it has been repeatedly done from the International Space Station (ISS). Most obviously, this right can only be executed by *remote* e-voting. There is just no alternative.[4]

Some 30 years after the science fictional introduction of remote e-voting within an e-society, it became science. Mathematically introduced after the breakthrough of asymmetric cryptography such as RSA [93] and ElGamal [40], together with the theoretical works in anonymizing mix-networks and blind signatures by David Chaum[22, 25] and the works of Josh Benaloh (formerly Cohen) in e-voting protocols [10] it became real when in 1992 Fujioka, Okamoto and Ohta gave birth to a first recognized remote e-voting scheme [45] comprising the emerging Internet.

[4]By the time writing this thesis, the hurricane 'Sandy' made landfall at the east coast of North America just days before the US-elections, hitting the state of New Jersey so badly, that the Christie Administration Announced: "E-Mail and Fax Voting Available to New Jerseyans Displaced by Hurricane Sandy". This shows that some better way of remote e-voting system is urgently needed, as e-mail was never intended to serve as a democracy maintaining e-voting variant.

The scheme was implemented by Votopia [66], an e-voting system used to vote for the "most valuable players" at the 2002 FIFA world cup.

But how is it that even 30 years after remote e-voting became a subject of science, there still is no e-voting scheme that reaches a level of common acceptance, whereas systems for remote e-banking are well established?

Why E-Voting is not E-Banking Confronted with the topic of remote e-voting for the first time, one is tempted to recognize a problem equivalent to remote e-banking. However, this equivalence is not given. In the following, an intuitive approach is chosen to demonstrate the fundamental difference.

In e-banking, there are three main parties involved: the customer, the electronic messenger and the bank. In accordance to the works of Michael Schläpfer, the electronic messenger comprises not only the public network (Internet), but also all of the computer equipment on the customers' side. The adversary model chosen for the worst case scenario is able to observe and alter the conversation at any time, completely replacing the electronic messenger (think of it as the worst hacker using the worst Trojan-Horse imaginable). The customer is not able to hide any private key material from the adversary, if the material is stored on or accessible by the computer. In this way the customer and the bank always communicate directly with the electronically omnipresent adversary. In that scenario, the adversary wins if it is possible to benefit from the conversation without being noticed by the other parties. Even though the practical chances for the adversary are quite attractive, in theory they are not. This is due to the possibility of the bank to watch a potential transaction involving a particular customer and hence to verify a transaction by communicating with the customer directly via an alternative channel not controllable by the adversary. In particular, the bank is able to ask the customer to confirm a transaction and the customer is able to raise a complaint against a specific transaction. This can be done via phone or even in person, or – if all the cryptographic assumptions hold – via a separate secure platform as presented by Weigold et al [115].

The ability to link every potential transaction to a specific customer at any time, allows the design of a verifiable e-banking protocol with the predicate "theoretically secure".

In e-voting, however, no party not even the voters themselves should ever be able to link a specific vote, influencing the tally, back to its creator. And in its strictest way, no other party but the voters themselves should have knowledge about the voter's participation and the voter's volition for a specific voting event. Being able to tear information from the voting system, resulting in knowledge of "if" or of "how" a certain voter voted, is considered as breaking the e-voting system. These requirements void the possibility of the e-voting authority to keep

track of every potential transaction per e-voter, and thus eliminates the alternative communication-channel to the voter by definition. As a result, this opens the doors wide for the adversary sketched above. The adversary wins if the final tally of an e-voting event can be altered unnoticed.

There is yet another subtle difference: In e-banking, neither the bank nor the customer are obliged to keep a relationship if there is a loss of faith on either side. In this case, there is no need to challenge the e-banking protocol any further. In the case of e-voting, however, the opposite is true. If the e-voting protocol fails to remain trustworthy at any moment, it does not fail for a single voter, but for an entire society, though blood might be shed.

Conclusively, remote e-voting requires an equal quality of verifiability as that provided by e-banking, but additionally has to respect democracy and everlasting privacy comprising voter-anonymity and the secrecy of the vote.

1.3. Contribution

During the past four years, I have been allowed to assist the metaphoric bullet extraction process and thus was able to contribute towards the above mentioned conclusion. In this sense, this thesis describes the extraction of three bullets.

1.3.1. Prepublications

The following results marking the cornerstones of this theses have been published as a joint work in several peer-reviewed international conferences and workshops or have been published in peer-reviewed journals and books:

International Workshop on Electronic Voting: (EVOTE'10)
Reto E. Koenig, Eric Dubuis, Rolf Haenni: Why Public Registration Boards are Required in E-Voting Systems Based on Threshold Blind Signature Protocols [68]

International Conference on E-Voting and Identity: (Vote-ID 2011)
Michael Schläpfer, Rolf Haenni, Reto E. Koenig, Oliver Spycher: Efficient Vote Authorization in Coercion-Resistant Internet Voting [99]
Oliver Spycher, Melanie Volkamer, Reto E. Koenig: Transparency and Technical Measures to Establish Trust in Norwegian Internet Voting [109]

IFIP International Information Security Conference: (SEC'11)
Reto E. Koenig, Rolf Haenni, Stephan Fischli: Preventing Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes [69]

International Conference on Financial Cryptography: (FC'11)
Oliver Spycher, Reto E. Koenig, Rolf Haenni, Michael Schläpfer: A New Approach towards Coercion-Resistant Remote E-Voting in Linear Time [110]

International Workshop on Electronic Voting: (EVOTE'12)
Oliver Spycher, Reto E. Koenig, Rolf Haenni: Achieving Meaningful Efficiency in Coercion-Resistant, Verifiable Internet Voting [107]

Journal of Computers & Security: (2013)
Rolf Haenni, Reto E. Koenig: A Generic Approach to Prevent Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes [56]

Design, Development, and Use of Secure Electronic Voting Systems: (Book chapter accepted for publication 2013)
Rolf Haenni, Reto E. Koenig, Eric Dubuis: Voting over the Internet on an Insecure Platform [54]

The following results have been presented on a workshop or platform without being peer-reviewed:

Dagstuhl Reports
Reto E. Koenig: How to Store some Secrets [67]

IACR Cryptology ePrint Archive 2012 375:
Reto E. Koenig, Rolf Haenni: How to Store some Secrets [70]

Studie im Aufrag der Schweizerischen Bundeskanzlei (BK'12)
Eric Dubuis, Rolf Haenni, Reto E. Koenig: Konzept und Implikationen eines verifizierbaren Vote Électronique Systems [38]

1.3.2. Contributions within Context

In 2005 Juels, Catalano and Jackobson introduced the first accepted coercion-resistant E2E-verifiable remote e-voting protocol [63] with respect to a very strict voter model. It has become famous under the abbreviation *JCJ05*. In theory, it allows democratic voting over the Internet in a coercive environment, in a well defined adversary model. However, this protocol is still far below the minimal

requirements (Figure 1.1(a)) of a remote e-voting system to be used for political elections and clearly has its shortcomings, as can be seen in Figure 1.1(b). All contributions but the last directly target improvements to JCJ05.

Contribution I: Bringing Practical Efficiency to JCJ05-Based Schemes

It is a well known and widely discussed fact that the original JCJ05-scheme has a major problem: It is prone to fail even under normal conditions. The tallying phase of JCJ05 cannot guarantee practical efficiency due to the unrestricted input and the quadratic computational complexity for processing it in order to get to the final tally. Hence, depending on the amount of legitimate input, the scheme cannot guarantee to be able to calculate the final tally within a reasonable time horizon (due to polynomial bounded computing power). Hence, the system cannot guarantee to reflect the sum of every legitimated vote coming from members of the electorate must be reflected in the final tally. But this is a fundamental requirement of democratic voting.

Rendering JCJ05 Linear

This work extends the approaches to reduce the computational complexity of JCJ05 from quadratic to linear computing time by maintaining the property of non-coercibility. This makes the scheme more efficient, as the computation of the final tally is easier to achieve. However, it seems inherent that lowering the computational complexity also lowers the degree of coercion-resistance (Figure 1.1(c)). Moreover, linearization approaches tend to bring more complexity into the scheme, so its realization becomes more difficult to achieve. The work described within this part builds upon previous approaches to this topic and in one case re-enables an approach that has been found to be broken within the original publication. This contribution is reflected within the following peer reviewed and published articles: [99, 107, 108]

Rendering JCJ05-Schemes Board Flooding Resistant

This work finally introduces practical efficiency at scheme level by eliminating the seemingly inherent weakness of a board flooding attack, bearing the imminent risk of not being able to calculate the final tally within a reasonable time horizon. The introduction of practical efficiency is done by allowing to set a hard upper bound with respect to computing time and space and thus eliminating the unrestricted amount of input. However, this work again comes with a negative impact on coercion-resistance. But, it is shown that the theoretical loss of coercion-resistance is minimal in practice. Even though this work improves usability it is still very poor with respect to human voters and asks for refinement. (Figure 1.1(c)) Consequently, a linear-time derivative scheme with very

little overhead is presented, combining the findings in this field of study. This contribution has been peer reviewed in the following published articles: [56, 69]

Contribution II: Managing High Entropy Credentials in the Context of Coercion

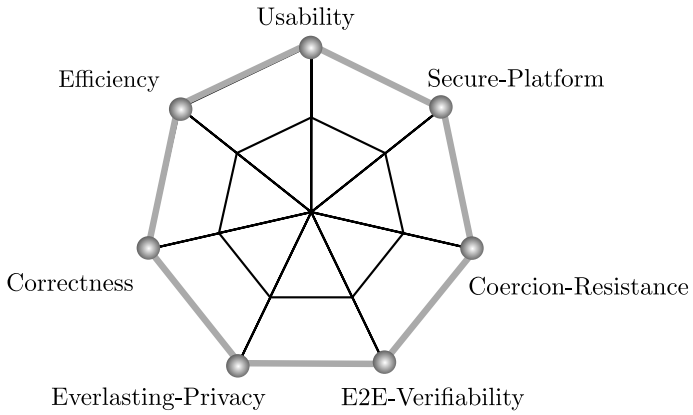
The poor usability provided within contribution I asks for a solution in order to allow humans to participate in the e-voting process. Thus, the second achievement masters the requirement of cryptographic protocols within the field of coercion-resistance, urging the human user to manage and discriminate multiple independent high entropy credentials obliviously even to a very strong adversary. A solution to this exact requirement has been provided by the introduction of a new cryptographic key-management scheme. It allows the management of multiple independent messages or credentials by passwords obliviously within a single ciphertext, hence representing a perfect hiding, deniable credential-management system within the field of coercion-resistant e-voting (Figure 1.1(d)). It comes with the special property of "no-search" [111], as the access to the desired credential only requires the entry of the corresponding key. A study of a user friendly implementation of this credential-management demonstrates the proof of concept in terms of usability and security [77]. The study shows a reduction in correctness 2.1 that comes naturally with the inherent habit of humans using low(er) entropy keys. But, in combination with an unchallengeable e-voting system, this problem stays on the theoretical side. This contribution has been published in the following articles: [67, 70]

Contribution III: Secure and Private Voting on Adversarial Ground

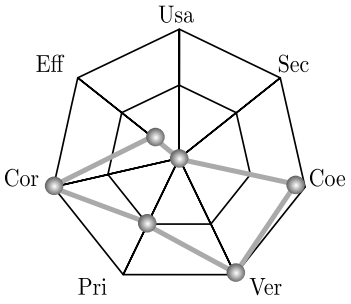
A concept study concerning a solution to the secure platform problem [94] presents a user interface allowing the management of most complex election scenarios on the voter's side on a user-friendly desktop computer—a highly untrustworthy environment. Integrity, secrecy and privacy of the vote is established with the help of an offline voting device requiring only minimal trust assumptions and allowed only restricted computing power. It bears the literal ability to constantly *observe* potential votes created by the voter on the desktop computer. Its ability to interact directly with the voter, unbeknownst to the desktop computer, enables the voter to detect an active adversary, to deceive a passive adversary and of course to vote in privacy (Figure 1.1(e)). A study [88] of an implementation of this concept proofs the unification of usability and strong privacy. This contribution states an extension of the following articles: [38, 54]

1.4. Structure

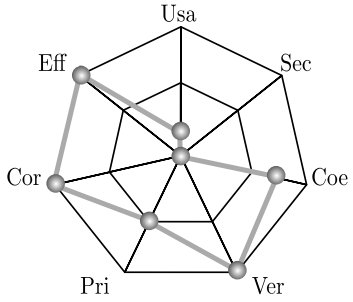
This thesis is structured as follows: Chapter 2 gives a detailed description of the field of operation, introducing the properties for democratic voting and the cryptographic tool chain required in order to defy the also described manifold adversarial models and their attacks against democracy. Chapter 3 provides a tight overview of the different classes of e-voting schemes, where a small contribution will show an attack on democracy in one of the fundamental schemes—declassifying their usage. Then the introduced contributions will follow, all of them covered within a separate part. The last part presents a fusion of the contributions and holds the conclusion of this dissertation.



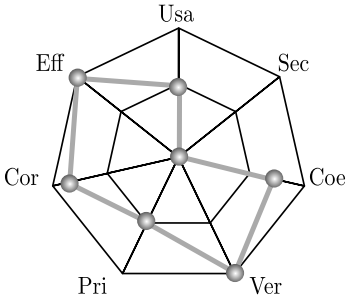
(a) Perfect System: The look of a democratic remote e-voting system fulfilling all minimal requirements in perfection



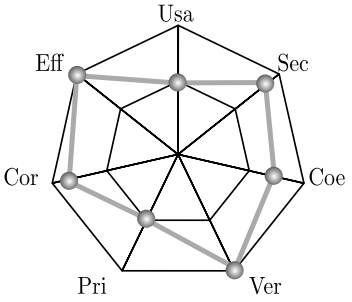
(b) JCJ05: Introduction of E2E-verifiability and coercion-resistance preserving correctnes



(c) Part$I$: Introduction of availability also brings usability but affects coercion-resistance



(d) PartII: Augmenting usability reduces correctness



(e) PartIII: Introduction of a secure-platform covers an untouched dimension within coercion resistant schemes

Figure 1.1.: The spider diagram represents 7 of the minimal qualities required for a trustworthy democratic e-voting protocol. To this day, even these 7 dimensions cannot be satisfied, or even quantified.

Chapter 2

Foundations of E-Voting

The aim of this chapter is the introduction of three fundamental models used in order to describe e-voting: *democracy*, the *adversary* and *cryptography*.

2.1. Democratic Voting

In the following, the foundation of democratic voting in general is discussed. The term *democracy* was introduced by the ancient Greeks and the true meaning of $\Delta\eta\mu o\kappa\rho\alpha\tau\iota\alpha$ is $\delta\eta\mu o\varsigma$ meaning "people" interpreted as "electorate" and $\kappa\rho\alpha\tau o\varsigma$ meaning "power". Hence democracy is the power of the electorate in order to make a decision about an object or a subject, where the *electorate* is defined as "all people entitled to participate in a particular event".

As humans show a natural addiction to power ($\kappa\rho\alpha\tau o\varsigma$), the individual electoral participant – *the voter* – has to be prevented from being influenced by power in all its shapes.

2.2. Properties of E-Voting

The prevention from power mentioned in the last section raises the following requirements for any voting scheme in focus within this thesis, expressed here with just enough precision so the requirements can be applied to the mathematical models within the domain of e-voting.

2.2.1. Correctness

The final tally of a voting event must express the volition of the electorate. This requires the scheme to present the following properties:

Democracy: Only the electorate is allowed to cast votes and each member of the electorate is allowed one single vote cast expressing the member's volition.

Integrity: No vote cast expressing a member's volition can be altered deleted or substituted.

Accuracy: Only and all votes cast expressing the member's volition are expressed in the final tally.

2.2.2. Privacy

The quintessence of privacy within e-voting is to provide the ability to freely cast the volition uninfluenced by the act of voting itself. Though, the only source where electoral information for any voting event is extractable is the final tally. This requires the scheme to feature the following properties:

Fairness: No individual or electoral information is extractable from the votes cast prior to the final tally.

Secrecy: Apart from the final tally, no further individual or electoral information is ever extractable from the voting scheme for any voting event.

Receipt-Freeness: This property combines and enhances fairness and secrecy in such a way that even the author of a vote cannot prove, willingly or unwillingly, the content of the vote which partly constitutes the volition expressed in the final tally [9].

As these properties cannot hold for an electorate comprising less than four members, the size of the electorate must exceed a certain amount of individuals in order to generate an *anonymity set*. The existence of such a set guarantees that no absolute volition information of an individual residing within the anonymity set can ever be extracted.

It is important to note that these properties forming privacy must remain true not only during the voting event, but forever. This last requirement is known under the term *everlasting privacy* [83].

2.2.3. Verifiability

As already mentioned in the introduction, the verification of the process ultimately must provide confidence to the voter that the vote was *counted as intended* and that the final tally has been formed under the democratic aspects. This confidence can be achieved by verifying the sub-processes, namely: cast as intended, recorded as cast, counted as recorded.

The more the processes of the voting scheme are verifiable, the less unconditional trust in the scheme – thus the final tally – is required. Trust and power are an unhealthy combination for all parties involved, including the trustees themselves. Verifiability comes in the following two qualities:

Individual Verifiability: Each member of the electorate is able to verify if the cast vote carrying their volition has been counted correctly in the final tally (*counted as intended*). Due to the requirements of privacy, this can only be done in a designated way. Hence the knowledge gained of this verification cannot be transferred to someone else.

Universal Verifiability: Everyone is able to verify the correctness of the final tally after a given voting event (*counted as recorded*). The gained knowledge of this verification is transferable to everybody. Note, that within this dissertation this definition also covers the special quality of *Eligibility Verifiability* allowing to verify publicly if the property of accuracy is respected.

It is important to note that these qualities of verifiability must be applicable forever, without ever diluting the required properties of privacy.

2.2.4. Robustness

Voting describes a special instance of a multi-party computation, where multiple different actors or parties calculate a consolidated result.[58] This way no minor group of individuals within such a party of the voting system in use (election authority, electorate, tallier, verifier, etc.) must be able to disrupt the properties of the scheme and thus compromise the voting event; neither on purpose nor accidentally. All of the above anticipates no single point of failure. Therefore it is required to distribute the duties of a party, by intrinsic redundancy, to guarantee its function up to a certain threshold. Moreover, no external group should be able to disrupt the properties of the voting system in use and thus compromise the voting event. This requires extrinsic redundancy of the parties involved in the voting event and a robust voting scheme, anticipating no weak point on the protocol level.

2.2.5. Access-Control

Democratic voting of any kind deals with a central problem: *access-control* [82]. This might come as a surprise, but in the end its a reduction to the question of who is allowed to access what data at which time. Thus, access-control is defined by two processes: *authentication* and *authorization*. Only if these two processes work perfectly together, access-control can prevail against any adversarial penetration.

As made clear in the dissertation of Mercury [82] access-control is not a physical or mathematical problem but a purely sociological problem. The locks and keys only support the processes of authentication and authorization, but provide no solution to the problem of access-control. This comes in accordance to a citation of Bruce Schneier: "If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology."

2.3. Adversary

Any party (individual or group) trying to benefit from breaking one or several requirements given within the field of operation, hence circumventing access-control, is considered an adversary. In the field of e-voting, the adversary attacks one of the properties mentioned in Section 2.1.

This thesis will always deal with a monolithic adversary [32] meaning that the there is only one instance of the adversary (maybe controlling other adversaries of its kind). Furthermore, the adversary is probabilistic and strict polynomially-time bounded [51], meaning that the adversary can use probabilistic (randomized) algorithms running on a polynomial-time bounded equivalent of a Turing machine. The adversary's knowledge may vary, depending on the context of operation. Even though the adversary may be modelled as mobile in space and time, it is always modeled in such a way that it never controls all entities of an e-voting system simultaneously.[50] Three main types of adversarial models can be discriminated in e-voting, namely *oblivious-, online-, offline-* adversary, representing different levels of knowledge:

Oblivious Adversary: This adversary knows the complete e-voting scheme and tries to attack on the protocol level using all publicly available information available at any time [21].

Online Adversary: In addition to the knowledge of the oblivious adversary, this adversary has access to all private data used for e-voting except the private keying material [21].

Offline Adversary: In addition to the knowledge of the online adversary, this adversary also has access to the private keying material used [21].

In the following, a new adversary model has to be introduced, which describes the real adversarial threat the most. Todays malware is able to extract private key material from the computer-equipment used by the victim, by putting a key-logger in action when the user types in the password to extract the private key. So, only if a remote e-voting system prevails against this new adversary model, it is worth discussing the property of coercion-resistance.

Definition 2.3.1 (Pseudo Offline Adversary) *The pseudo offline adversary obliviously controls all computers stating a universal Turing machine (a freely programmable computer) used by the victim, and has access to all information including the private keying material if it is processed on a controlled computer. However, this adversary explicitly has no access to any trusted* limited *computing device, whereas* limited *is defined as not stating a universal Turing machine.*

This way the victim can hide information by processing it on a trusted limited computing device.

Furthermore, the adversary can be modeled with different behavior:

- *Passive versus Active*: A *passive adversary* only observes the victim, meaning that the victim is still able to act according to the e-voting scheme, whereas the *active adversary* is able to take control over the victim [32].

- *Static versus Adaptive*: A *static adversary* chooses an immutable set of victims prior to the attack, whereas an *adaptive adversary* can alter the set of victims before and during the attack [32].

It must be noted that in any combination the victim does not have to be directly aware of the presence of any adversary at all.

2.4. Cryptography

The main approach chosen in remote e-voting for access-control, in order to maintain its minimal properties, is the usage of cryptography. The reasoning seems quite logic, as the primary working field of cryptography has always been politics and the protection of knowledge in order to maintain or gain power. Cryptography started its modern renaissance with the upcoming personal computer in the 1980-ies. After its release from pure military classification in the late 1990-ies (an era also known as the *Crypto Wars* [19]) it made its way into everyday life in modern e-society. But as already mentioned: cryptography does not state a solution but only acts as a helping mechanism for access-control.

In contrast to the physical world, in the digital world cryptographers need to provide quite an arsenal in order to reach the same supportive protection (locks, keys, transparent ballot boxes, seals, talliers, observers) against the described adversary. Hence, a set of specialized *cryptographic primitives* is required to develop *cryptosystems* representing the building-blocks of today's attempts within the field of e-voting. This section covers the introduction to these building blocks on an abstract level. By introducing a notation independent of any concrete implementation, the

described building-blocks can be used in the forthcoming sections without the risk of mathematical entanglements.

Apart from standard cryptosystems for encryption, decryption, signature and hashing, the toolbox used for e-voting comprises some more complex building blocks, such as non-interactive zero-knowledge proofs of knowledge, verifiable secret-sharing, verifiable threshold cryptosystems, verifiable plaintext equivalence tests, verifiable re-encryption mix-nets, verifiable exponentiation mix-nets, anonymous channels, and the append-only public bulletin board. In the following these building-blocks and their purposes are introduced based on the definitions and notations used by the books "Handbook of Applied Cryptography" [81] and "Introduction to Modern Cryptography" [64].

2.4.1. Encryption and Decryption

Every cryptosystem is defined by three probabilistic polynomial-time algorithms: *Gen, Enc, Dec* whereas these algorithms perform the following:

Gen: Key generation takes as input the security parameter n and outputs a key $k \in \mathcal{K}$ of the according *key space*. Key generation must be polynomial time depending only on the security parameter. The security parameter n represents the amount of algorithmic tries in order to find any key without further knowledge, and is described as an unary initialisation string taken from 1^n. (e.g. 1^n is equivalent to a string represented by n bits). For deterministic key generation, some additional secret input parameter $k' \in_R \{0,1\}^*$ is required, where R indicates the random tape.

Enc: Encryption takes as input a message $m \in \mathcal{M}$ of the according *message space* and a key $k \in \mathcal{K}$ and outputs a ciphertext $c \in \mathcal{C}$ of the according *ciphertext-space*.

Dec: Decryption takes as input a ciphertext $c \in \mathcal{C}$ and a key $k \in \mathcal{K}$ and outputs a message $m \in \mathcal{M}$.

A cryptosystem must guarantee that $(\forall n, \forall k \in \mathcal{K}, \forall m \in \mathcal{M} | Dec_k(Enc_k(m)) = m)$, meaning that every cryptosystem is injective and allows efficient inversion from the *plaintext space* or *message space* $\mathcal{M}$ with $|\mathcal{M}| > 1$ into the *ciphertext space* $\mathcal{C}$ and back in polynomial time by applying according elements from the *key space* $\mathcal{K}$. Furthermore, the cryptosystem must be (t, ϵ)-secure, where $\epsilon = 2^{-n}$; meaning that an adversary can break a specific ciphertext c by finding m without prior knowledge of m and the complete keying material k with a probability of at most $t2^{-n}$ (where t is the number of tries) [64].

Symmetric-Key Encryption: In symmetric-key encryption also known as *private-key encryption*, a secret key $k \in \mathcal{K}$ has to be shared amongst the different parties enabling them to encrypt and decrypt messages. It is very important that the adversary does never gains access to k. This renders key distribution an act of trust[1].

Key generation: $k' \mapsto_n k$

This operation generates a secret key $k \in \mathcal{K}$ by a security parameter n and some secret random parameter k'. Please note that the forms of k' and k heavily depend on the mathematical context. $k = Gen_n(k')$.

Encryption: $m \mapsto_k c$

Maps a single message $m \in \mathcal{M}$ to a single ciphertext $c \in \mathcal{C}$ using $k \in \mathcal{K}$. It is expressed by the function $c = Enc_k(m)$.

Decryption: $c \mapsto_k m$

Maps a single ciphertext $c \in \mathcal{C}$ to a single message $m \in \mathcal{M}$ using a key $k \in \mathcal{K}$. If the same key $k_i \in K$ is used for decryption and encryption of the ciphertext c it will result in the same message $m_i \in M$. It is expressed by the function $m = Dec_k(c)$.

Multi-Encryption: This cryptographic concept defines a novelty which is formally introduced in Chapter 6.

In contrast to the previously described cryptosystem, a multi-encryption system maps a vector $M \in \mathcal{M}^n$ containing multiple messages $(m_1, \ldots, m_n)$ in a single ciphertext $c \in \mathcal{C}$ by applying the vector $K \in \mathcal{K}^{(n)}$ containing n individual, distinguishable keys $(k_1, \ldots, k_n)$.

Multi-encryption offers the special property of *partial decryption*: Each individual message $m_i \in M$ in c is exclusively and directly accessible by applying the according key $k_i \in K$ to the ciphertext.

Key generation: $K' \mapsto_n K$

This operation generates the key vector $K \in \mathcal{K}^{(n)}$ consisting of $(k_1, \ldots, k_n), k_i \neq k_j$for $i \neq j$ by a security parameter n and some secret parameters $K' = (k'_1, \ldots, k'_n)$, where $k' \in K' \subseteq \{0,1\}^*$.The forms of k' and k heavily depend on the mathematical context. The complete function for key generation is expressed by $K = Gen_n(K')$.

[1] It is not enough to rely on a secure channel during key distribution. It is important that every party holding a copy of the key remains trustworthy as long as messages have to remain secret which might be everlasting. This in turn requires unlimited trust in all involved parties without knowing the future of the relationship amongst those parties; hence, this is also a sociological problem.

Encryption: $M^n \mapsto_K c$

Maps n-messages $m_1, \ldots, m_n \in \mathcal{M}$ to a single ciphertext $c \in \mathcal{C}$ when a key vector K containing n distinct keys $k_1, \ldots, k_n \in \mathcal{K}$ is applied. It is expressed by the function $c = Enc_K(M)$.

Decryption: $c \mapsto_{k_i} m$

Maps a single ciphertext $c \in \mathcal{C}$ to a single message $m \in \mathcal{M}$ when the corresponding key $k_i \in \mathcal{K}$ is applied. It is expressed by the function $m = Dec_k(c)$.

Asymmetric-Key Encryption: In contrast to symmetric cryptosystems, where the same key k is used for encryption and decryption, in asymmetric-key encryption also known as *public-key encryption*, the key k consists of two distinct parts, a private part x drawn from the *private-key space* $\mathcal{X}$ used for decryption and its public opponent y, part of the *public-key space* $\mathcal{Y}$ used for encryption. This results in $\mathcal{K} = \mathcal{X} \times \mathcal{Y}$ and thus in the requirement $(\forall n, \forall (x, y) \in \mathcal{K}, \forall m \in \mathcal{M} | Dec_x(Enc_y(m)) = m)$. In public-key encryption the requirement of the system to be (t, ϵ)-secure also comprises the finding of the private key x for a given ciphertext c and the according public key y. In contrast to key distribution in the symmetric setup, this time the public-key y can be distributed without any trust assumption from the key-distributors side.[2]

Key generation: $x' \mapsto_n (x, y)$

This operation generates the key $k \in \mathcal{K}$ consisting of the tuple $(x, y) \in \mathcal{X} \times \mathcal{Y}$ by a security parameter n and some usually ephemeral secret parameter x', whereas $x' \in \{0, 1\}^*$. In contrast to other encryption systems the key (x, y) cannot be chosen freely, as the key parts strictly depend on each other. This usually results in high entropy key parts not easy to remember for the human brain. Please note, that as in the former case, the forms of x', x and y heavily depend on the mathematical context. The complete function for key generation is expressed by $(x, y) = Gen_n(x')$.

Encryption: $m \mapsto_y c$

Maps a single message $m \in \mathcal{M}$ to a single ciphertext $c \in \mathcal{C}$ when a public key $y \in \mathcal{Y}$ is applied. It is expressed by the function $c = Enc_y(m)$.

Decryption: $c \mapsto_x m$

Maps a single cipher $c \in \mathcal{C}$ to a single message $m \in \mathcal{M}$ when the corresponding private key $x \in \mathcal{X}$ is applied. It is expressed by the function $m = Dec_x(c)$.

[2]However, the sociological problem remains, as the message sender using the public-key for encryption still has to trust the receiver of the message, holding the secret key part.

Special Properties of Encryption Schemes: In the following, some special properties of some of these cryptosystems will be described. These properties are of great value in the field of e-voting.

Homomorphic Encryption: $Enc_y(m_1) \otimes Enc_y(m_2) = Enc_y(m_1 \odot m_2)$

Allows the application of an operation $\otimes$ within the ciphertext space $\mathcal{C}$ to be equivalent to the application of an operation $\odot$ within the plaintext space $\mathcal{M}$. Hence, an encryption is said to be homomorphic in respect to addition, if the ciphertext operation $Enc_y(m_1) \otimes Enc_y(m_2)$ is equal to $Enc_y(m_1 + m_2)$ whereas + defines the arithmetic addition. And so an encryption is called homomorphic in respect to multiplication, if the ciphertext operation $Enc_y(m_1) \otimes Enc_y(m_2)$ is equal to $Enc_y(m_1 \times m_2)$ whereas $\times$ defines the arithmetic multiplication.

The encryption is called *fully homomorphic* if there exist two operations within the ciphertext space, one resulting in a multiplication within the plaintext space and the other in an addition within the plaintext space [47].

Any homomorphic property renders the encryption malleable, meaning that even if the plaintext representation of a specific ciphertext c is not known, one can change the contained plaintext at one's will. For example, in an asymetric setup, it is possible to double the content of an encryption within a multiplicative homomorphic encryption system by performing a homomorphic operation on the ciphertext without further knowledge of the plaintext or the keying material.

Probabilistic Encryption: $(m, r) \mapsto_y c$

Allows the encryption of a message $m \in \mathcal{M}$ to different ciphertexts c, c' using different *randomization* $r, r' \in_R \mathcal{R}$, where r and r' represent ephemeral random elements of the *randomization space*. Hence, the encryption function requires an additional parameter for the randomization and is expressed as $c = Enc_y(m, r)$. However, decryption remains untouched and thus results in m for c as well as for c' without explicit knowledge of r or r'.

Re-Encryption: $(c, r) \mapsto c'$

An encryption cryptosystem where the homomorphic property (for the operation with the identity element) is preserved under probabilistic encryption also possesses the natural property of *re-encryption*. This property allows changes in the representation of the ciphertext, whereas the representation of the plaintext remains unchanged $Enc_y(m) = c, ReEnc_y(c) = c', c \neq c', Dec_x(c) = Dec_x(c')$. In public-key encryptions the function of re-encryption can be applied without further knowledge of the private keying material or the plaintext of the ciphertext c , but by using the same public key y as was used for the creation of c. It is expressed as $c' = ReEnc_y(c, r)$.

2.4.2. Hash Functions

In principle, a hash function h maps elements of a set $\mathcal{D}$ of arbitrary size (could be infinite) to a finite range $\mathcal{R}$. Elements of $\mathcal{R}$ and elements of $\mathcal{D}$ are also referred as the *image* and the *preimage*. However, in cryptography, things are a bit trickier than that. Even worse, there exist many different definitions of a cryptographically secure hash function. This thesis will follow the definitions given in the "Handbook of Applied Cryptography" [81] and hence the following properties must hold in order to call a hash function a cryptographically secure, *collision resistant* hash function:

Hash: $x \mapsto_h y$
: Maps a domain element (preimage) $x \in \mathcal{D}$ to an image $y \in \mathcal{R}$ using the function h.

Collision Resistance: $h(x) \neq h(x'), x \neq x'$
: It must be computationally infeasible to find two distinct domain elements $x, x' \in \mathcal{D}$ such that $h(x) = h(x')$. This further implies the following properties:

  Preimage Resistance: $h^{-1}(y) = ?$
  : Given an image $y \in \mathcal{R}$ it must be computationally infeasible to find a potential domain element (preimage) $x \in \mathcal{D}$ such that $y = h(x)$. Hence the hash function cannot be inverted for any y.

  Second Preimage Resistance: $h(x) = y, h^{-1}(y) = ?$
  : Given a domain element (preimage) $x \in \mathcal{D}$ and its corresponding image $y \in \mathcal{R}$ it must be computationally infeasible to find another preimage x' such that $h(x') = y$.

As the authors in [30] mention, the term "*computationally infeasible* to find" is fuzzy in this context. Here it means that no function has been found so far that does the trick. Hence, a cryptographically secure, collision resistant hash function is defined as a one-way function without any (known) trapdoor. The function of the just described *secure cryptographic hash* is expressed by $h = Hash(m)$.

2.4.3. Message Validation

The main purpose of message validation is the proof of authorship by a remote trusted party. This is especially important for messages transmitted over an asynchronous, insecure channel. It allows the receiving party to verify whether the message received has been approved by the claimed remote party and thus, that the message sent by the remote party has not been altered during transmission. Every such validation system allows a mapping from the *message space* $\mathcal{M}$ into the *signature space* $\mathcal{A}$ and a mapping from $\mathcal{C}$ to a boolean set by applying elements from the *key space* $\mathcal{K}$.

Message Authentication: The main purpose of message authentication is the designated proof of message approval for a predetermined set of trusted parties. As in private-key encryption, it is very important that the adversary does not gain access to the verification key k. The proof is not transferable to any third party. This results from the fact that any party in possession of k is able to execute the signature function for any data, whereas parties not in possession of k are not able to execute the verification function. Thus, this message validation scheme is of type "private-key" [50].

Digital Signature: In contrast to message authentication, the main purpose of a digital signature is the universal proof of message authorship. As in public-key encryption, the key is divided into a pair (s, v), a secret one for signing and the other one for verification, which can be made publicly available even to the adversary. Hence this message validation scheme is of type "public-key" [50]. As is done in encryption systems, the signature is required to be (t, ϵ)-secure. As a consequence, providing enough security to the digital signatures makes the proof unforgeable without knowledge of the private signing key. This brings forth the special property of non-repudiation for an approved message; once a message has been signed, the signer cannot deny the approval of that particular message.

Key generation: $s' \mapsto_n (s, v)$

This operation generates the tuple(s, v) by a security parameter n and some usually ephemeral secret parameter s'. For message authentication $s = v$ might be true[3]. Please note, that the forms of s', s and v heavily depend on the mathematical context. The complete function for key generation is expressed by $(s, v) = Gen_n(s')$.

Sign: $m \mapsto_s a$

Maps a message $m \in M$ to an authorship verification element $a \in A$ when the private signature key $s \in S$ is applied. It is expressed by the function $a = Sig_s(m)$.

Verification: $(a, m) \mapsto_v boolean$

Maps an authorship verification element $a \in A$ to a boolean value when the corresponding public verification key $v \in V$ is applied. This also works for message authentication. However, in that case v has to be kept secret from the adversary. It is expressed by the function $boolean = Ver_v(a, m)$.

rSign: $(m, r) \mapsto_s a$

Maps a message $m \in M$ to different authorship verification element a, a' using

[3] e.g. RSA-Signature-Scheme

different *randomization* $r, r' \in_R \mathcal{R}$, where r and r' represent ephemeral random elements of the *randomization space* $\mathcal{R}$ of the signature function. Hence, the signature function requires an additional parameter for the randomization and is expressed as $a = Sig_s(m, r)$. However, verification remains untouched $(m, r') \mapsto_s a', a \neq a', r \neq r'$ and thus results in the same verification for a as well as for a' without explicit knowledge of r or r'.

Blind Signature: Blind signatures allow the approval of a message without knowing its content [23]. They are of great use when a message needs to be signed without disclosing it directly to the signer. The idea behind blind signature schemes is to render a message eligible by a trusted third party. This is done by blinding the message m to $\tilde{m}$ prior to its signature using an ephemeral secret *blinding factor* b. After signature, the blinding has to be reverted on $\tilde{m}$ as well as on the blinded signature $\tilde{a}$ in order to get m and s by applying the *unblinding factor* b^{-1}. As before, a blind signature must at least be (t, ϵ) – *secure*, meaning that an adversary can break a specific blinding factor b with a probability of at most $t\epsilon$. Three additional functions are needed for a blind signature:

Blinding-factor-generation: $b' \mapsto_n (b, b^{-1})$
This operation generates the tuple (b, b^{-1}) for blinding and for unblinding by a security parameter n and some usually ephemeral secret parameter b'. Please note that the forms of b', b and b^{-1} heavily depend on the mathematical context. The complete function for key generation is expressed by $(b, b^{-1}) = Gen_n(b')$.

Blind: $m \mapsto_b \tilde{m}$
Just prior to requesting a signature on a message m by the signing party, m is blinded by applying the blinding factor b to it. It is expressed by the function $\tilde{m} = Blind_b(m)$.

Unblind: $(\tilde{m}, \tilde{a}) \mapsto_{b^{-1}} (m, a)$
After the reception of the signed blinded message $\tilde{m}$, m as well as a, have to be unblinded, so the signature a will approve the original message m. It is expressed by the function $(m, a) = Unblind_{b^{-1}}(\tilde{m}, \tilde{a})$.

2.4.4. Zero-Knowledge Proof of Knowledge

A zero-knowledge proof of knowledge (*ZK-PoK*) allows a party (the *prover*) to demonstrate to another party (the *verifier*) that a mathematical statement is true, without revealing anything other than the truth of the statement itself. This demonstration is done in several steps establishing a protocol. All e-voting schemes

in focus within this thesis only require a special instance of ZK-PoK protocols, based on homomorphisms within a finite domain (e.g., exponentiation homomorphism with known order co-domain). These special ZK-PoK's are all covered by the Schnorr-protocol, usually described within a Σ^{ϕ}-*protocol*, whereas Σ indicates the message transfer between the prover and the verifier, and ϕ denotes the homomorphism in use [34, 42, 91, 100]. The following description is based on the dissertation of Krenn [72] and Maurer [79].

Σ^{ϕ}-Protocol: Let $\phi : \mathcal{G} \mapsto \mathcal{H}$ be a homomorphism from the group $\mathcal{G}$ to the group $\mathcal{H}$, such that $|\mathcal{G}| < \infty$ and let $y = \phi(w)$. Providing a challenge set $\mathcal{C} = \{0, \ldots, c^{+} - 1\}, c^{+} = |\mathcal{G}|$, a common input (y, ϕ) and the prover's private input w, the Schnorr-protocol Σ^{ϕ} is defined as a two-party protocol (P, V), where the verifier V is a probabilistic polynomial-time algorithms (PPT) and the prover $\mathrm{P} = (\mathrm{P}_1, \mathrm{P}_2)$ is a pair of algorithms:

- P_1 draws $r \in_R \mathcal{G}$, and outputs $t = \phi(r)$ and keeps r secret.
- Upon receiving the challenge $c \in_R \mathcal{C}$, P_2 outputs $s = r + cw$.
- V outputs accept if and only if $ty^c = \phi(s)$

There are different ways in order to get the challenge c. Using the interactive 3-move protocol Σ^{ϕ}-protocol illustrated in Figure 2.1(a), a honest verifier is required. However, in practice, a non interactive 1-move protocol is in use where c is received by a publicly known cryptographic hash function $c = H(t|y)$ [4]. This procedure is referred to as Fiat-Shamir heuristic[42]. See Figure 2.1(b) for an illustrated protocol flow. This thesis will remain on the 1-move protocol (NI-ZK-PoK). Appart from providing zero knowledge this form must provide the following properties:

Completeness: Completness is given by $\phi(r) \times \phi(wc) = \phi(r + cw)$, hence, for honest P, V accepts.

Special Soundness: With overwhelming probability a dishonest P without knowledge of w is not able to present an accepting proof. [5]

Zero Knowledge: No additional knowledge than the proof of knowledge of P can be gained from an accepting protocol run.

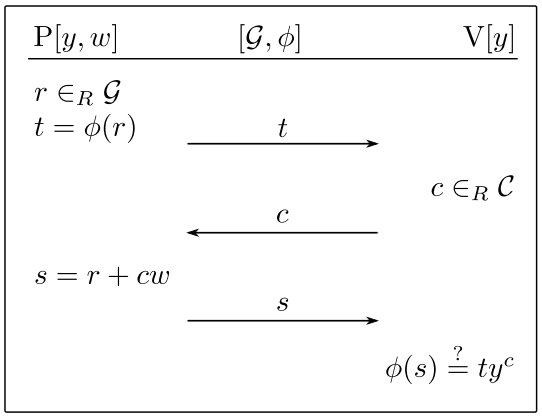
[4] Note that taking y into account for hashing is only required if y is not common knowledge (Existing PKI), but is only known publicly when communicating the proof [12]

[5] Soundness is only given in the Random Oracle Model, Hence, if the hash function in use would behave like a true random function. However, in practice a weaker but publicly accepted cryptographic hash algorithm such as SHA-2 is in use.[112]

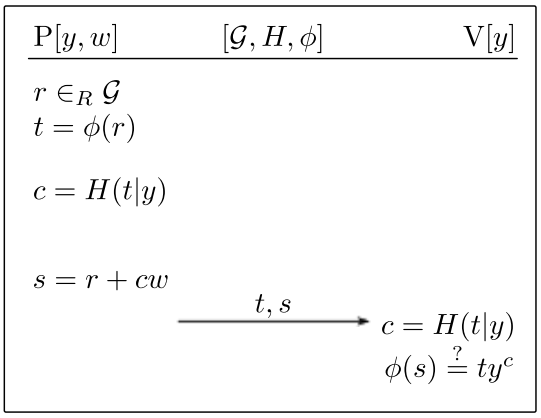
A NI-ZK-PoK comes in two contexts established by the prover:

Universal: If the prover establishes the NI-ZK-PoK in this context, the proof is verifiable to everyone—universaly. This is the default context of the used NI-ZK-PoK.

Designated: If the prover establishes the NI-ZK-PoK in this context, the proof only enables the predefined verifier in focus to infer the correctnes of a statement [62]. This can be achieved by an 'or' composition of two statements in one proof: Either the verifier knows the witness (t) or some private data only known to the verifier. Given a public key setup, the private key of the verifier can be used. This way, the accepting proof will only carry information for the verifier in focus. Hence it is designated.



(a) ZK-PoK:
The structure of an interactive 3-move Schnorr (Σ)-protocol



(b) Ni-ZK-PoK:
The structure of a non-interactive 1-move Schnorr-protocol with Fiat-Shamir heuristic

Figure 2.1.: Different variants of the Schnorr-protocol

Notation: In the following, the standard notation for (NI)ZK-PoKs for homomorphisms is introduced allowing to hide their inner complexity [6, 20, 79].

Elementary Proof of Knowledge: The construction of an elementary proof of knowledge π is expressed by $\pi = ZKP\{(w) : y = \phi(w)\}$.
The and composition of two homomorphisms with a common co-domain is expressed by $\pi = ZKP\{(w_1, w_2) : y_1 = \phi_1(w_1) \wedge y_2 = \phi_2(w_2)\}$.

Compositional Proof of Knowledge:
For the sake of simplicity the conjunctive and disjunctive compositions of two proofs π_1, and π_2 are expressed by $\pi_1 \wedge \pi_2$, and $\pi_1 \vee \pi_2$. However, these simple expressions may harbour complex constructions.

2.4.5. Verifiable Mix-Nets

A mix-net consists of a sequence of mixing servers, $M_1, \ldots, M_s$, each receiving a batch of input messages and producing a batch of output messages in a permuted (mixed) order [25]. In addition to providing privacy by unlinking output from input messages, mix-nets must guarantee the integrity of the processed messages. Therefore zero-knowledge proofs of correct mixing must be published by each mixing server along with the generated output messages [52, 61, 84, 90, 97, 116].

There are various types of mix-nets [98] but within this thesis only two are of interest.

Re-Encryption Mix-Net: In a *re-encryption mix-net*, the input messages are encryptions $c_i = Enc_y(m_i, r)$ and the output message are corresponding re-encryptions $c'_{\psi(i)} = ReEnc_y(c_i, r')$, where $\psi = \psi_s \circ \cdots \circ \psi_1$ and $r' = r'_1 + \cdots + r'_s$[6] are compositions of permutations ψ_i and randomizations r'_i, respectively, selected at random by each M_i. Generally, if ψ and r' remain secret, then the adversary cannot determine which output message corresponds to which input message. Clearly, a single mixing server not colluding with the adversary is sufficient for this to hold. However, if all but one mixing server unveil their ψ or r, the honest mixing server gains the complete knowledge in a passive way and hence privacy is broken. So in order to guarantee privacy, at least two mixing server need to keep their ψ and r secret.

Exponentiation Mix-Net: *exponentiation mix-net* creates a blinded permutation of an input batch, where all entries are permutated and each entry is blinded by a common value α (an exponent in this case). Hence, using a permutation ψ_i and a secret α_i selected at random, each mixer permutates and transforms input messages X_i into blinded output messages $X'_{\psi(i)} = X_i^{\alpha_i}$ [55].

2.4.6. Secret-Sharing

Secret-sharing allows to split a secret message into multiple shares and distribute these shares among independent secret holders. The message can only be recovered if

[6] in case where r' are additive elements as for example in an ElGamal [40] setup.

the shares are combined again. There exist several secret-sharing schemes. However, within this context, only Shamir's scheme [104] is of interest. The sharing is called *threshold* if the message can be recovered combining only t-out-of-n shares, where t is the predefined minimum (the threshold) amount and n represents the total amount of shares distributed. The combination of shares in order to unveil the secret message is expressed by the function $m = Combine_t(s_1 \dots s_t)$

Trusted Dealer: If the message and the secret shares are created by some party, the secret-sharing is established by a trusted dealer. This implies that at least the dealer is in exclusive possession of the complete message. The creation is expressed by the function $\mathcal{S}_m = Share_{(t,n)}(m)$

Distributed Generation: It is possible to create shares representing an unmanaged and hence unknown message that will only be known if the shares are combined. This way, distributed secret holders jointly generate the message by committing themselves to a share per holder. If their commitment is accompanied by a zero knowledge proof of the commitment the secret-sharing is publicly verifiable and the possibility of cheating by a minority of secret holders is excluded. The creation is expressed by the function $\mathcal{S}_m^v = Share_{(t,n)}()$

2.4.7. Threshold Cryptosystem

A public-key cryptosystem is called a threshold cryptosystem if the private key x is shared among n parties, and if the decryption can be performed only by a threshold number of parties $t \leq n$ without explicitly reconstructing x and without disclosing any information about the key shares. A threshold version of such a public-key cryptosystem results from sharing the private-key x using distributed secret-sharing. The *distributed decryption* allows the ciphertext to be decrypted by applying the portions of the secret shares, ultimately unveiling the encrypted message m without the need of explicitly reconstructing x. This implies that no party will ever learn about the private-key.

2.4.8. Plaintext Equivalence Test

A plaintext equivalence test (PET) allows to verify if two ciphertexts contain the same plaintext, without revealing any other information about the plaintexts. Easy in deterministic encryptions, it is also possible in probabilistic homomorphic threshold cryptosystems and is expressed by the function $\{true, false\} = PET_y(c_1, c_2)$.[60]

2.4.9. Anonymous Channel

An anonymous channel hides the correspondence between senders and their messages, i.e., the senders of the messages remain anonymous or untraceable. The most common realization of anonymous channels is based on *onion routing*. One of today's most widely used implementations of an anonymous Internet channel is Tor [37, 49] Users of a secure Tor can, if properly set up, hide their identity while browsing the Web. As an alternative to using such designated systems, people may protect the privacy of their online activities by accessing the Internet from public access points (Internet cafés, public libraries, public WLAN, etc.), although the anonymity provided in that case may not always be perfect.

2.4.10. Untappable Channel

In fact, an untappable channel is not a cryptographic element, but rather a physical requirement. It requires that there be no evidence whatsoever of any communication that has taken place between two entities, no matter how much computing power the adversary can bring in. It is the equivalent of the perfect voting booth (required only during the supervised voting setup). However, there are efforts to replace the physical requirement by using deniable encryptions [59].

2.4.11. Public Bulletin Board

A public bulletin board (*PBB*), sometimes called a web bulletin board [57], allows the user to publish data, guaranteeing that it cannot be altered in any fashion without being noticed. As stated in the master thesis of Beuchat [13], a secure public bulletin board must present the following requirements: It must be *available* for authorized users in order to publish messages and for everybody to be able to read the consistent content of the board. It must guarantee an *unalterable history* so no data whatsoever can be altered once made available. It must provide *no single point of failure* implying that a PBB must be implemented as a distributed system with threshold, where a minority of components may fail without any consequences for the proper functioning of the PBB. And, it must be *verifiable* so that the user can verify if the PBB respects the properties just mentioned. However, in 2000 Eric Brewer [14] formulated a theorem called *CAP*-theorem stating the impossibility of a distributed computer system to provide *C*onsistency, *A*vailability and *P*artition tolerance perfectly at the same time. This conjecture was proven in 2002 by Gilbert et al [48]. This renders the true implementation of an *optimal* PBB a very delicate task itself and thus no practical robust implementation is at hand so far.

2.5. Secure Platform Problem

Even though it seems as if the odds for cryptography are quite good in order to provide support for maintaining democracy in e-voting, the following problem hit the e-voting community by ambush. The *secure platform problem*, introduced by Ronald Rivest in the year 2001 [94], describes the following dilemma in cryptography: As the mathematical and information theoretical requirements of secure cryptosystems surmount the abilities of the human brain, cryptographic activities have to be executed using a computer. However, it is not possible to prove, that no adversarial software is running on the users computer. This is due to the fact that this problem is equal to the Halting-Problem, which is known to be theoretically undecidable on a Turing machine. As the computers in question are equal to universal Turing machines (except the bounded space), this observation states that the existence of adversarial material of any kind on the computer cannot be blamed on the user. This leads to the fact that a secure e-voting system must be designed in such a way, that it accepts the presence of the previously described, extremely powerful pseudo-offline adversary.

So, this seems to be the premature end of remote e-voting, and this thesis should come to an abrupt end. But wait: The halting problem becomes theoretically decidable for a device which is not Turing complete (hence not freely re-programmable). This implies the theoretical proof of the absence of adversarial material and hence voids the holistic threat of the pseudo-offline adversary model. Please note, that this only remains true if all operations involving the use of the private key are executed on this *limited device*. As such, it is still much more powerful than the human brain when it comes to cryptographic computations and can serve as the missing link between the requirements of cryptography and the potential of the human brain. This implies that each voter requires a physical device, easy enough to handle and analyze, strong enough to support cryptographic needs, and fulfilling the minimal requirements described for the first time in the 'Orange Book' in 1985 [35]. These observations give the remaining parts of this theses the legitimation of existence. However, it is not the only decisive for the outcome of the e-voting debate described in the introduction.

Chapter 3

Remote E-Voting Overview

This chapter chronologically introduces schemes representing cornerstones of the path towards coercion-resistant E2E-verifiable remote e-voting. They are divided into the following artificial categories, namely historical, multiparty-computation and coercion-resistant; each category builds on the earlier ones. Every scheme is introduced by its original title followed by a short overview of the cryptographic components involved and a brief description of the core functionality. Finally the scheme in focus is checked against the core requirements for democratic voting.

3.1. Classical Remote E-Voting Scheme

This section gives an introduction to how an e-voting scheme is defined and described. In the following, it gives an introduction on how to analyze an e-voting scheme in order to get a feeling of how subtle the subject is:

FOO92: A Practical Secret Voting Scheme for Large Scale Elections
In 1992 Fujioka et al. presented the following scheme, depending on blind signatures. [45] In addition to the administrator, it consists of three entities, namely the voter, the eligibility authority and the collector authority.

Initialization: The administrator creates a public list of the electorate by publishing the voters and their electoral identities (Tokens).

Setup / Voting: The purpose of this phase is to authorize eligible voters to cast their votes. The voter creates a commitment $m = Enc_k(v)$ (symmetric encryption) of the vote v, then sends the blinded commitment $\tilde{m} = Blind_b(m)$ together with the electoral identity id to the eligibility authority. The eligibility authority checks if the voter is indeed eligible and whether the voter

has not yet requested to vote. If both checks are positive, it signs the blinded commitment $\tilde{a} = Sign_a(\tilde{m})$ and sends it back to the voter. At the end of this phase, the eligibility authority publishes a list S of tuples $(\tilde{a}, id)$. Meanwhile, the voter unblinds the signed commitment $a = Unblind_{b^{-1}}(\tilde{a})$ and is now ready for vote casting.

Casting: The voter casts the tuple (m, a) representing the commitment of the vote and the corresponding signature to the collector-authority. The sending is done via an anonymous channel. The collector-authority collects and verifies all the signed commitments. It publishes all positively verified commitments on a public bulletin board.

Tallying: The tallying process requires all voters to decommit, hence to decrypt their published vote. This is done by sending the key k via an anonymous channel to the collector. After the opening of the vote, the tallying process can gather all the plaintext votes and compile the final tally.

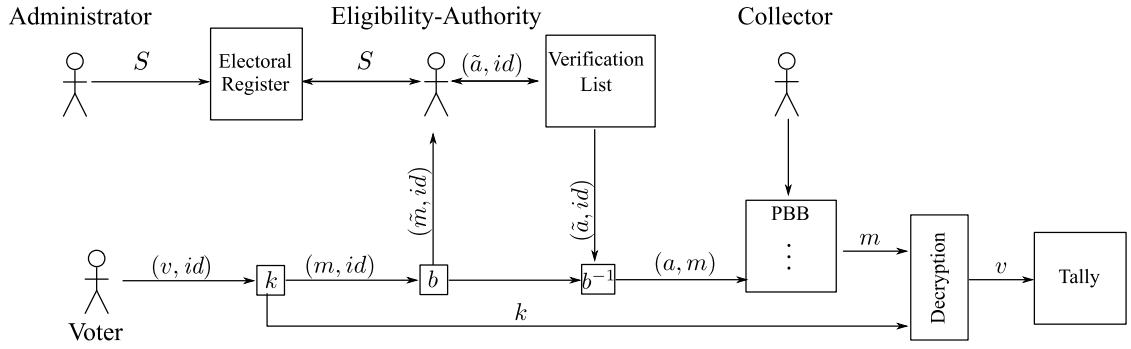


Figure 3.1.: Overview of the FOO92-scheme: Each voter gets their legitimization via a blind signature. Vote casting is done in two consecutive steps: presenting the committed vote and decommitting.

Correctness

Democracy: Is not given, as the eligibility-authority is able to silently start *ballot stuffing* by signing its own votes and handling them as votes coming from voters of the electorate not participating on the event. This attack is not universally verifiable. The same is true if the eligibility-authority is colluding with the administrator. And in any case, the administrator is always able to create fake members of the electorate, hence breaking democracy without being noticed, neither universally nor individually.

Integrity: Is given.

Accuracy: Is not given, as the scheme requires each voter to access the voting system twice. The problem arises if the voter only manages to access the voting system in the first step, but is not able to provide the decryption key in the second step. In this case, a voter's volition is available but not decryptable. This problem can only be solved if the complete casting is done within a single step known under the term *vote and go*[1].

Privacy

Fairness: In its weakest sense, this property is given, However, the system universally leaks information about all voters not having participated on a given voting event.

Secrecy: Is given.

Receipt-Freeness: Is not given, as the voter is able to universally prove knowledge of the private key in order to match the commitment and the plaintext vote.

Veriviability

Individual Verifiability: Is given, as each voter can verify if the (blindly-)signed encrypted vote truly represents the voter's intention.

Universal Verifiability: Is not given, as one cannot verify if only eligible voters have cast their votes (eligibility verifiability).

Robustness

This protocol does not show any attempt in this direction. The only entity that

[1]The property of vote and go has been introduced in the concrete e-voting system called Votopia [66] already mentioned in the introduction of this thesis. Votopia presents a variant of FOO92

is replicated is the entity covering the voters. But, neither the administrator nor the eligibility authority and the collector are replicated by any means. The protocol fails should one of these entities fail. Failing can have different causes: Not able to work properly or not willing to work properly.

3.2. E-Voting Schemes with Respect to Secure Multi-Party Computation

In order to achieve a certain amount of robustness, remote e-voting has to be introduced to the field of secure multi-party computation with threshold. To avoid an entity becoming a single point of failure, the entity is replicated n times where it is assumed that at least a certain minimum amount t of replicates work in the sense of the protocol.

3.2.1. Breaking Blind Signature Schemes in Secure Multi-Party Computation Setup

Within a (n, t)-threshold environment, the eligibility-authority described in the setup phase of FOO92 can be replicated to n independent authorities, such that at least t different signatures are required in order to be authorized for casting the vote. This environment allows to satisfy the requirement of fairness, as no eligibility-authority is able to gain knowledge on the voters not having participated in a voting event. Moreover democracy can be hardened as long as at least t eligibility authorities are needed in order to start ballot stuffing. Even though this distribution of power and knowledge seems to be a good solution in general, it bears some nifty attacks, as could be demonstrated in [68]. Within this small contribution an attack on democracy is described, when threshold blind signature schemes are in use: Emmanuel Benoit[2] discovered that a group of colluding voters can ultimately cast additional votes without being detected, and thus violate the requirement of democracy. The following example presented in [68] illustrates the attack:

- available eligibility-authorities: $n = 4$
- authority signature threshold: $t = 3$

A *fair* voter v_i generates four blinded messages (one blinded message $\tilde{m}_{ji} = Blind_{bj}(m_i)$ per authority A_j, $1 \leq j \leq 4$) containing the same commitment m_i:

[2] A member of the team at the RISIS-Institute

	A_1	A_2	A_3	A_4
v_i	$\tilde{m}_{1i}$	$\tilde{m}_{2i}$	$\tilde{m}_{3i}$	$\tilde{m}_{4i}$

Each authority signs the blinded message and returns the signature $\tilde{a}_{ji}$ to the voter. To cast the vote, the voter sends the message together with three out of four unblinded signatures a_{ji} anonymously to the collectors. The voter discards the remaining signature.

A *malicious voter group* consisting of three colluding voters v_u, v_v, v_w of the electorate, where $u \neq v \neq w \neq u$, can acquire an additional casting authorization $\tilde{m}_x$ resulting in four independent vote casts:

	A_1	A_2	A_3	A_4
v_u	$\tilde{m}_{1u}$	$\tilde{m}_{2u}$	$\tilde{m}_{3u}$	$\tilde{m}_{4x}$
v_v	$\tilde{m}_{1v}$	$\tilde{m}_{2v}$	$\tilde{m}_{3x}$	$\tilde{m}_{4v}$
v_w	$\tilde{m}_{1w}$	$\tilde{m}_{2x}$	$\tilde{m}_{3w}$	$\tilde{m}_{4w}$

The following holds true:

- $\tilde{m}_u$ is rendered valid by the signatures a_{1u}, a_{2u}, a_{3u} of authorities A_1, A_2, A_3;
- $\tilde{m}_v$ is rendered valid by the signatures a_{1v}, a_{2v}, a_{4v} of authorities A_1, A_2, A_4;
- $\tilde{m}_w$ is rendered valid by the signatures a_{1w}, a_{3w}, a_{4w} of authorities A_1, A_3, A_4;
- $\tilde{m}_x$ is rendered valid by the signatures a_{2x}, a_{3x}, a_{4x} of authorities A_2, A_3, A_4.

This is possible as the different registration authorities operate independently from each other and, thus, no synchronization takes place amongst them. Even though the attack is only possible at a linear scale in terms of colluding voters, it is still significant. The quantitative impact of the attack is proportional to the number of colluding voters.

Due to the nature of threshold there always exists a subset of size $n-t$ authorities not needed in order to get sufficient signatures for a valid vote cast. Let V_c be the size of a malicious colluding voter group. Hence, the maximum number of additional vote casts v_+ that can be rendered valid by the malicious voter group, is:

$$v_+ = \lfloor \frac{n-t}{t} V_c \rfloor$$

For the threshold values n, t such that $\frac{2}{3}n \leq t \leq \frac{3}{4}n$, v_+ is in the range of:

$$\frac{V_c}{3} \leq v_+ \leq \frac{V_c}{2}$$

The violation of democracy shown above is present in all protocols based on threshold blind signature, where the blinding procedure results in a different message for every individual signer.

Therefore, a common registration board (PBB) must be used as a knowledge base for synchronization amongst the registration authorities. However, this requirement is contradictory to the requirement of fairness, as one can estimate the amount of participating voters at any given time during the voting phase.

The demonstration of this attack led our group to stop any further study on threshold blind signatures within the e-voting environment.

3.2.2. CGS97: A Secure and Optimally Efficient Multi-Authority Election Scheme

In 1997 Cramer et al. [33] introduced a new class of remote e-voting schemes in the secure multi-party computation setup. The basic idea of this scheme is to use the homomorphic properties of a probabilistic threshold cryptosystem in conjunction with zero-knowledge proofs. As mentioned previously, the threshold cryptosystem allows decrypting operations without explicitly reconstructing the private key (Section 2.4).

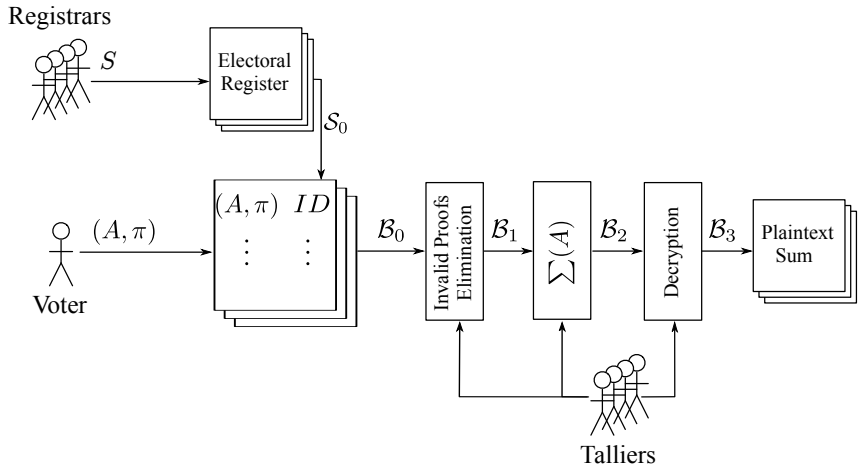


Figure 3.2.: Overview of the CGS97-scheme: The vote is placed beside the voter's ID on the public bulletin board. The filter eliminates votes with invalid proofs, the remaining votes are summed up, and the sum is decrypted to get the final result.

Given the assumption that a finite set $\mathcal{C}$ of available choices in a given referendum or election is publicly known, the voter's actual choice is written as $c \in \mathcal{C}$. The

choices are represented in such a way that they can be summed up arithmetically in order to allow the calculation of the final tally. A valid ballot consisting of an encrypted vote $A = Enc_y(c)$ and the corresponding non-interactive zero-knowledge proof

$$\pi = ZKP\{(r_A) : \bigvee_{c \in \mathcal{C}} A = Enc_y(v, r_A)\},$$

to prove that A represents an encryption of $c \in C$. After the voting period, all ballots of eligible voters form the set $\mathcal{B}_0$. After removal of all ballots with invalid proofs, the remaining ballots form the set $\mathcal{B}_1$, which is ready to be counted. The sum of all encrypted votes form the singleton $\mathcal{B}_2$. Finally the element in $\mathcal{B}_2$ is jointly decrypted by the authorities $\mathcal{B}_3$ If a threshold amount of authorities does not collude, the votes themselves will never be decrypted. As indicated with S_0 in Figure 3.2, this schema allows to present the identity of the corresponding voter beside the encrypted vote without breaking privacy in terms of secrecy.

Correctness

Democracy: Is given as the identity of the voter resides beside every encrypted vote cast. This way the electorate can be verified by anyone. Even though in general an impersonation attack is not possible if each member of the electorate is in possession of a signature key within a public key infrastructure *PKI*, it can be detected at any time by the true member of the electorate and so this member can raise a complaint (re-vote).

Integrity: Is given via the PBB.

Accuracy: Is given by the PBB and the fact that the addition of the encrypted votes can be reproduced by anyone. The decryption of the sum is joined by a ZKP of correct decryption and is universally verifiable.

Privacy

Fairness: In its weakest sense, this property is given, However, the system universally leaks information about all voters not having participated on a given voting event.

Secrecy: Is given by the cryptographic assumptions.

Receipt-Freeness: Is not given by nature, as the voter's id is located beside the encrypted vote. As the voter is in possession of the randomization used to produce the probabilistic encryption, the voter can reproduce this

specific encryption at any time and the voter's will is thus available in plaintext.

Veriviability

Individual Verifiability: Is given, as each voter can verify if the encrypted vote truly represents the voter's intention (if the randomization of the encryption allows to decrypt the ciphertext).

Universal Verifiability: Is given, as one can verify if only eligible voters have cast their votes (eligibility verifiability) and one can verify if the cast votes have made it in the final tally.

Robustness

As this protocol is designed within the secure multi-party computation setup, it offers an adjustable degree of robustness.

An implementation of this scheme called *Helios* [2] has been used in multiple election events. Helios is open sourced[3] and is still evolving [31].

3.2.3. HS00: Efficient Receipt-Free Voting Based on Homomorphic Encryption

In 2000 Hirt et al. [58] presented the first truly receipt-free scheme by combining the CGS97 protocol with a re-encryption mix-net, containing n mixers and the designated-verifier proofs. In this scheme, the voter does not encrypt the vote, but the authorities have to provide a preprocessed set of all possible voting choices. Then, each of the n re-encryption mixers M_i mixes the set of choices N-times—once per member of the electorate. This way there are N individual permutation sets $\Psi_j = \{\psi_1, \ldots, \psi_n\}$ available, each of which has to be communicated to the corresponding member of the electorate v_j. This communication has to go over an untappable channel in order to render this scheme receipt-free—extremely heavy requirements are needed during each voting period and for every member of the electorate. This way, each voter possesses designated knowledge about the permutation of the individual voting choices and is thus able to point at the one representing the voter's volition. This pointing is done as in CGS97, by placing the voter's id beside a mixed vote.

[3] http://heliosvoting.org/

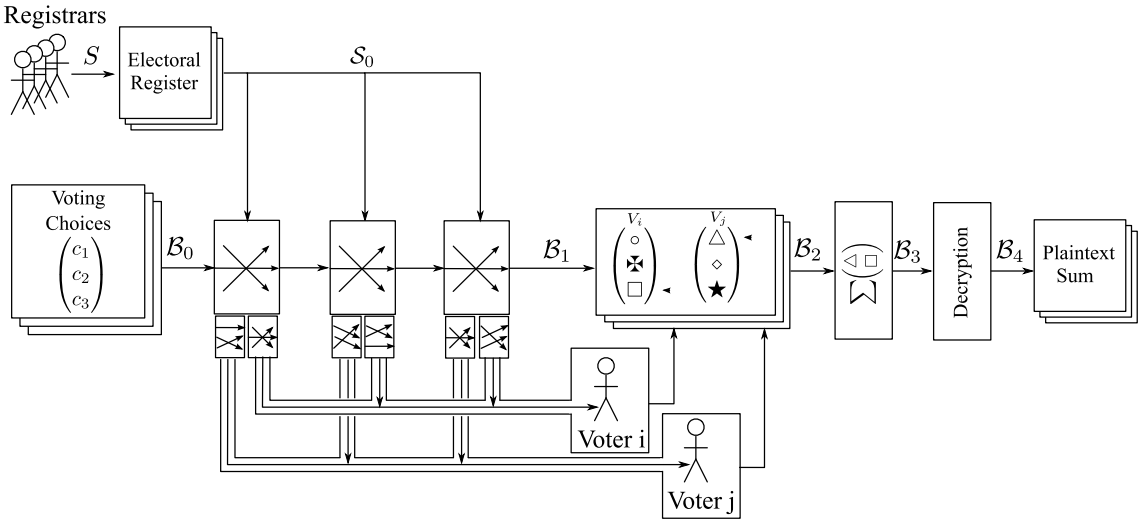


Figure 3.3.: Overview of the HS00-scheme: Each voter receives the permutations of the mix-nets via an untappable channel and thus is able to mark their volition beside the mixed and re-encrypted votes in $\mathcal{B}_1$. The marked votes are summed up, and the sum is decrypted to get the final result.

Correctness

Democracy: Is given as only the votes that are accompanied by the identity of a voter are summed up. This way the electorate can be verified by anyone. As in CGS97 an impersonation attack is not possible if each member of the electorate is in possession of a signature key within in a public key infrastructure *PKI*.

Integrity: Is given via the ZKPs of the verifiable re-encryption mix-net that are stored on the PBB.

Accuracy: Is given by the PBB and the fact that the addition of the encrypted and votes that have been pointed at can be reproduced by anyone. The decryption of the sum is joined by a ZKP of correct decryption, and is thus universally verifiable.

Privacy

Fairness: In its weakest sense, this property is given, as long as a threshold amount of the decrypting authorities act according to the scheme and as long as at least one mixer acts honestly. However, the system universally

leaks information about all voters not having participated in a given voting event.

Secrecy: Is given by the cryptographic assumptions.

Receipt-Freeness: In its weakest sense, this property is given, as the voter cannot universally prove the chosen option of the pointed vote. This is due to the fact that the untappable channel transmits receiver designated messages only. However, it is possible to demonstrate vote abstention, which might be regarded as a special form of a receipt.

Verifiability

Individual Verifiability Is given implicitly, as each voter gets the individual permutation matrix of the shuffled candidate order. Together with the ZKP of correct encryption and the ZKP of correct shuffling, every voter knows which candidate is represented by which ciphertext.

Universal Verifiability Is given, as one can verify if only eligible voters have cast their votes (eligibility verifiability), one can verify if all possible choices have been encrypted correctly (via ZKP), one can verify if all possible choices have been permuted correcty and one can verify if the cast votes have made it in the final tally.

Robustness

Again, as this protocol is designed within the secure multi-party computation setup, it offers an adjustable degree of robustness. This scheme allows to verify universally all aspects of correctness.

Even though the HS00-scheme offers receipt-freeness, the voter can still be attacked in various ways resulting in not being able to vote freely. Beside the already described *forced-abstention attack*, the voter can be forced to commit to a randomly chosen encrypted vote, resulting in a *randomization attack* described in [63]. Even though this case the attacker does not learn about the committed choice, citing the example provided in [63], "...an attacker favoring the Republican party in a United States election would benefit from mounting a randomization attack against voters in a heavily Democratic district".

3.3. E-Voting Schemes with Respect to Coercion-resistance

In 2005 Juels et al. formally introduced the term *coercion* as an attack model within the context of e-voting schemes, covering the above mentioned attacks. The following description presented in [56] recapitulates the various attack scenarios this model covers:

- In a *randomization attack*, voters are forced to vote for a random selection of candidates. The goal of this attack is to nullify with high probability the choice of the group of voters under attack, for example by selecting them from an area with a predictable election outcome. Note that for this attack to succeed, the attacker (and perhaps even the coerced voters) does not need to learn the actual candidate selection.

- In a *forced-abstention attack*, voters are forced to abstain from participating in the election, either by not casting a vote at all or by casting an invalid vote. With respect to its goal and effectiveness, this type of attack is closely related to a randomization attack, but much easier to achieve. By simply observing the public bulletin board in a scheme prone to this kind of attack, no direct interaction with the coerced voter is needed.

- In a *simulation attack*, voters are forced to hand over the legitimization to vote, for example by handing out the private voting credentials to the coercer, who can then impersonate (simulate) the coerced voters and thus vote on their behalf.

Any scheme allowing the voter to withstand this attack-model is considered to be *coercion-resistant*.

3.3.1. JCJ05: Coercion-Resistant Electronic Elections

In 2005 Juels et al. introduced the first coercion-resistant e-voting scheme [63]. Furthermore, they introduced the plaintext equivalence test (PET) (Section 2.4). Using this element, they handed back the creation of the vote to the voters themselves. This allows to minimize the use of the untappable channel to the registration phase only, providing a protocol with much more realistic assumptions and less unrealistic needs than requested by HS00.

Given the assumption that a finite set $\mathcal{C}$ of available choices in a given referendum or election is publicly known, the voter's actual choice is written as $c \in \mathcal{C}$ (a single option or candidate, a set of options or candidates, an ordered set of options or

candidates, etc.). The existence of a robust public bulletin board is required as a central public data storage. The encryption scheme used needs to be probabilistic and has to provide a homomorphic property and thus being re-encryptable (Section 2.4). A general overview of the scheme is shown in Figure 3.4.

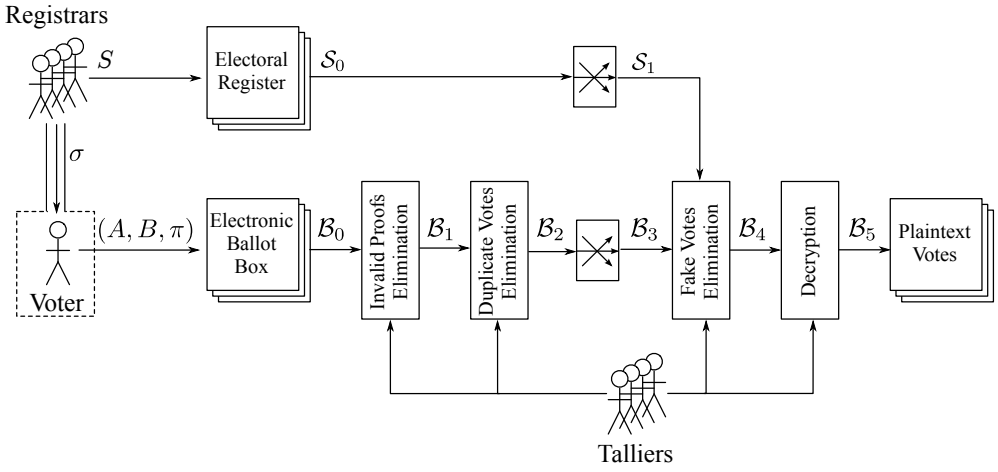


Figure 3.4.: Overview of the original JCJ05-scheme: the first filter eliminates votes with invalid proofs, the second filter eliminates duplicate votes, and the third filter eliminates fake votes by checking them against the electoral register.

Registration: The *registrars* jointly establish a random credential $\sigma \in \mathcal{M}$ where $\mathcal{M}$ defines the plaintext space[4] and deliver it to the voter via an untappable channel. Additionally, they jointly compute the encryption $S = Enc_y(\sigma, r_S)$, where y represents the talliers' common public key and r_S is a secret randomization. Finally, the registrars add S to the voter's entry in the *electoral register*, which resides on the public bulletin board. Assuming a threshold amount of trustworthy registrars, only the voter will know σ and no one will know r_S. At the end of the registration phase, the complete electoral register is digitally signed by the registrars. $\mathcal{S}_0$ denotes the set of all encrypted credentials in the electoral register, and $n = |\mathcal{S}_0|$ is the size of the electorate.

Vote Casting: The voter chooses $c \in \mathcal{C}$ from the set of available choices and computes encryptions $A = Enc_y(\sigma, r_A)$ and $B = Enc_y(c, r_B)$. To cast the vote,

[4]Using a concrete encryption-scheme, i.e. ElGamal, $\mathcal{M}$ states the set G_q, containing the quadratic residues of $\mathbb{Z}_p^*$ where $\mathbb{Z}_p^*$ defines a muliplicativ modular group with prime-modulus $p = 2q + 1$ and q is prime.

A and B must be accompanied by a conjunctive composition $\pi = \pi_A \wedge \pi_B$ of two non-interactive zero-knowledge proofs,

$$\pi_A = ZKP\{(\sigma, r_A) : A = Enc_y(\sigma, r_A)\},$$
$$\pi_B = ZKP\{(r_B) : \bigvee_{c \in \mathcal{C}} B = Enc_y(c, r_B)\},$$

one to prove knowledge of σ and one to prove $c \in \mathcal{C}$.[5] If the voter desires to fake a vote, σ can be replaced by any other element of the message space $\mathcal{M}$, a so-called *fake credential*. The resulting *ballot* (A, B, π) is posted to the *electronic ballot box* $\mathcal{B}_0$, which resides on the public bulletin board. $N = |\mathcal{B}_0|$ denotes the number of ballots in the electronic ballot box at the end of the vote casting phase.

Tallying: Five consecutive steps are necessary to detect and eliminate invalid ballots and to derive the election result from the remaining encrypted votes (see Figure 3.4). The main actors in the tallying phase are the *talliers*, which share the private decryption key x and jointly perform corresponding computations.

1. The proofs π are verified for all ballots $(A, B, \pi) \in \mathcal{B}_0$. Ballots for which the proof does not hold are excluded from further processing. The remaining reduced ballots (A, B) form a new set $\mathcal{B}_1$.

2. If multiple ballots contain the same plaintext credential, all but one of them are excluded from further processing according to some predefined election policy. This is the case if $PET(A, A') = \mathit{true}$ holds for two distinct ballots $(A, B) \in \mathcal{B}_1$ and $(A', B') \in \mathcal{B}_1$. The remaining ballots form a new set $\mathcal{B}_2$.

3. The sets $\mathcal{B}_2$ and $\mathcal{S}_0$ are mixed in two separate verifiable re-encryption mix-nets. These mix-nets produce two new sets $\mathcal{B}_3$ and $\mathcal{S}_1$.

4. Ballots not containing a valid credential are excluded from further processing. This is the case for a ballot $(A, B) \in \mathcal{B}_3$, if $PET(A, S) = \mathit{false}$ holds for every $S \in \mathcal{S}_1$. The remaining encrypted votes B form a new set $\mathcal{B}_4$.

5. The encrypted votes $B \in \mathcal{B}_4$ are jointly decrypted. This yields a new set $\mathcal{B}_5$, which contains the plaintext votes ready to be counted.

[5] The first proof π_A prevents attackers from casting unauthorized votes by re-encrypting entries from the electoral register (recall that r_S is not known to anyone). Since each authorized vote on the bulletin board will be decrypted during the tallying phase, π_B is needed to prevent coercers from forcing voters to select $c \notin \mathcal{C}$ according to some prescribed pattern, thus obtaining a receipt [36].

Correctness

Democracy: Is given as only the votes accompanied by a valid credential are counted, whereas a valid credential is defined as a credential provided to a member of the publicly known electorate. Furthermore, duplicate votes are eliminated, hence only one vote per eligible voter builds the final tally. The impersonation-attack is not possible as each member of the electorate gets the credential via the untappable channel. The credential cannot be forced from the voter, as the system allows the voter to hand out bogus[6] credentials. This fact also spoils the randomization-attack, as the coerced voter can simply use bogus credentials while forced to vote in random order.

Integrity: Is given as all cryptographic operations need to provide the corresponding universally verifiable ZKPs, which are stored on the PBB.

Accuracy: Is given by the PBB and the fact that the decryption of the votes is joined together with a ZKP of correct decryption and thus universally verifiable.

Privacy

Fairness: is given, as long as a threshold amount of the decrypting authorities act according to the scheme and as long as at least one mixer acts honestly. Due to the anonymous channel used within this scheme, no information about participating or non-participating members of the electorate is leaking.

Secrecy: is given by the cryptographic assumptions.

Receipt-Freeness: is given, as the voter cannot prove universally if the credential used is a valid credential.

Verifiability

Individual Verifiability Is given if and only if the voter perfectly manages the credential σ. This way, the voter knows if the vote will influence the final tally, as the system provides ZKPs for each processing step.

Universal Verifiability Is given, as one can verify if only eligible voters have cast votes influencing the final tally and that no eligible voter was able

[6] In this context, bogus is meant to be an invalid credential not known to the e-voting system.

to do so multiple times (eligibility verifiability), and one can verify if the cast votes of the eligible voters have made it in the final tally.

Robustness

Again, as this protocol is designed within the secure multi-party computation setup, it offers an adjustable degree of robustness.

Beside fulfilling all requirements of democratic voting, all defined coercive-attacks are ruled out. Seemingly, this protocol finally brings remote e-voting to the voter's PC. However, as of the time of writing (2012) there is no protocol implementing this scheme that is ready to cope with real elections. One main reason behind this fact lies in the amount of time such a protocol requires in order to tally the votes. In general, the complete transition from $\mathcal{B}_0$ to $\mathcal{B}_5$ runs in $\mathcal{O}(N^2 + N{\cdot}n)$ time.[7] This implies $\mathcal{O}(n^2)$ for $N \leq n$ and $\mathcal{O}(N^2)$ for $N \geq n$. The quadratic growth rate in both cases makes the scheme impractical in a large-scale setting.

[7] This is the asymptotic running time in terms of number of PETs. The number of mix servers and the size of the resulting proof π_B (which depends on $|\mathcal{C}|$) are not taken into account. A more detailed running time analysis is covered in [99].

Part I.

Bringing Practical Efficiency to JCJ05-Based Schemes

Chapter 4

Rendering JCJ05-Schemes Linear

Within this chapter, the problem of the inefficient tallying procedure of the original JCJ05 protocol is discussed and various promising protocol improvements with a linear-time tallying procedure are demonstrated.

4.1. JCJ-Based Schemes

All of the following approaches partly rehabilitate a technique allowing to do PET-equivalents in linear time. This technique has been introduced 2005 by Smith and Weber. Shortly after the introduction, this alternative had been found broken in their setting. In the following, the original improvements of Smith and Weber are described together with the attack that has broken the technique in general. Then a short summary of our approaches is given. Furthermore, some efficiency studies are presented.

4.1.1. Scheme by Smith and Weber

In 2006 Smith [105] and later Weber [113, 114] proposed the following: Instead of applying PET(A, A') pairwise on all elements of $\mathcal{B}_1$ for removing duplicates (Figure 3.4), they suggested to decrypt the messages (σ) after having blinded each with the same *blinding value* z. $A^z = Enc_y(\sigma, r_A)^z, Dec_x(A^z) = \sigma^z$ The blinding value z is a composition of random values (from within the mathematical environment the public encryption scheme is operating in) shared among the m talliers $z = \sum_{i=1}^{m} z_i \in_R$. The resulting blinded values (σ^z) are then stored in a hash table. This approach allows a collision-detection in linear time without additional information leakage (see Figure 4.1).

.

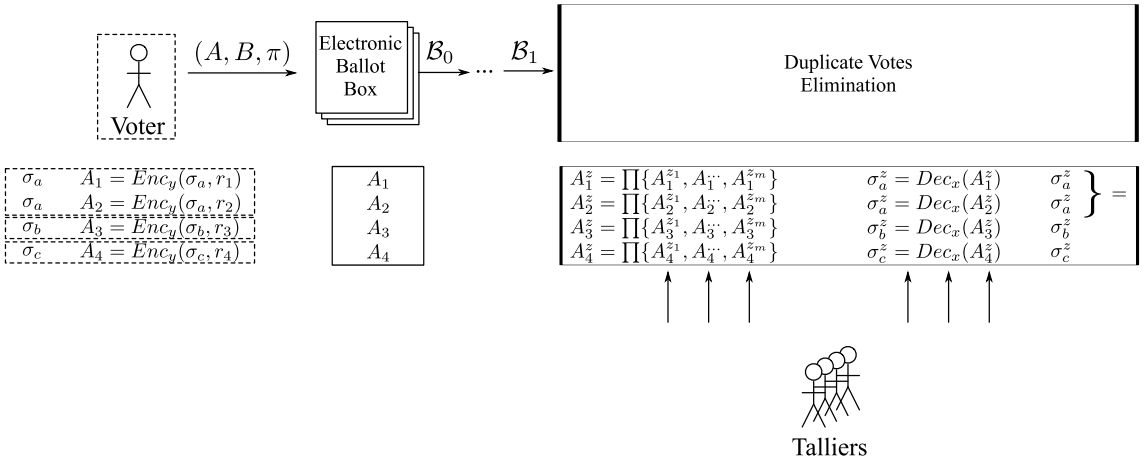


Figure 4.1.: Example of a Smith and Weber approach for duplicate elimination within the JCJ05 scheme. Four encrypted messages are sent to the electronic ballot box, whereas the messages A_1 and A_2 contain the same value (σ_a). After the blinding and decryption, these duplicates are detected.

However, both authors proposed using the same procedure again for eliminating fake votes (at this point of the scheme, a mix-net has been used to break the relationship of the voter and the vote). In this case however, the coercer gets an attack strategy enabling the re-establishment of the link from a vote to a specific voter. This ultimately allows the coercer to identify whether a vote with known σ makes it into the final tally. [3, 29, 89]

4.1.2. SHKS11: Scheme by Schlaepfer, Haenni, Koenig, Spycher

The basic idea behind the SHKS11 scheme [11, 99] is that each voter indicates the correct location of the publicly encrypted credential behind an anonymity-set (see Figure 4.2).

The registration step is identical to the original JCJ05-scheme. In addition to computing values A and B along with a conjunctive proof $\pi = \pi_A \wedge \pi_B$, a subset $I \subseteq \{1, \ldots, n\}$ of size β is chosen at random and added to the ballot. This is the ballot's anonymity set, which must include the voter's own index i. After excluding ballots with an invalid proof, duplicate votes (As) are removed as in Smith's and Weber's schemes. For every remaining ballot (A, B, I, π), the talliers create β new ballots (A, B, S) by retrieving S from the electoral register at every index $i \in I$. The resulting list of ballots is mixed in a re-encryption mix-net. Finally, the talliers

perform a single PET for each ballot. If $PET(A, S)$ returns *true*, B is decrypted and counted. The complete tallying procedure runs in $\mathcal{O}(\beta N)$ time.

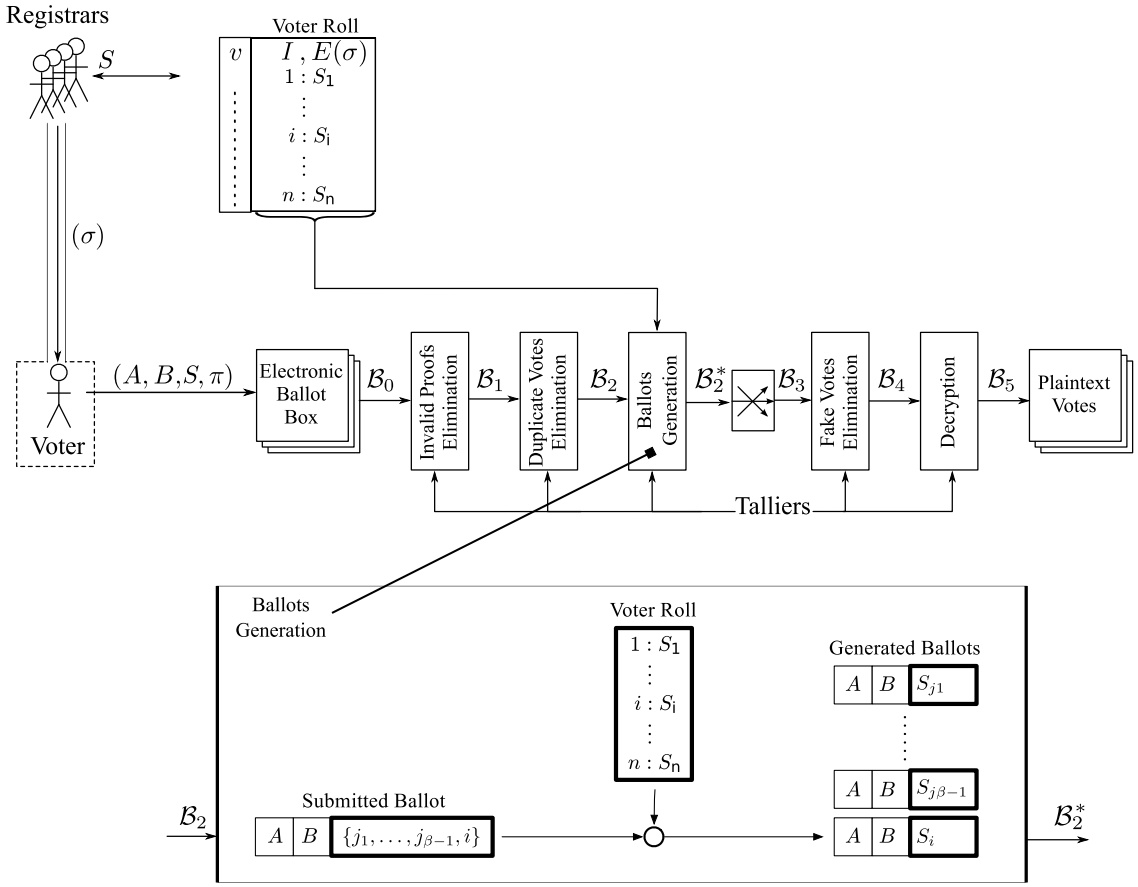


Figure 4.2.: Overview of the SHKS11 Scheme: As the voter sends the indices of the possible encrypted σs, the fake-vote elimination becomes linear. Please note, that the ballot generation process just before the mix-net is illustrated in order to explain that step in more detail.

4.1.3. SKHS11: Scheme by Spycher, Koenig, Haenni, Schlaepfer

The basic idea behind the SKHS11 scheme [108] is that the voter indicates the correct location of the publicly encrypted credential encrypted for the talliers. The talliers then introduce an anonymity set for each voter.

The registration step is conducted according to the original JCJ05-scheme. In addition to values A and B, the voter computes $C = Enc_y(i, r_C)$, where i is the index

of the voter's entry in the electoral register. In the extended proof $\pi = \pi_A \wedge \pi_B \wedge \pi_C$, π_A and π_B are as in the original scheme and $\pi_C = ZKP\{(i, r_C) : C = Enc_y(i, r_C)\}$ proves knowledge of i. The resulting ballot (A, B, C, π) is posted to the electronic ballot box. After excluding ballots with an invalid proof, the talliers generate a random number (β in the average) of additional fake votes for each index i. After removing duplicate votes (As) as in Smith's and Weber's schemes, the resulting list of ballots (A, B, C) is mixed in a first re-encryption mixnet. Next, the talliers jointly decrypt C into i and establish a new set of ballots (A, B, S) by retrieving S from the electoral register at index i. This set is mixed in a second re-encryption mixnet. Finally, the talliers perform a single PET for each ballot. If $PET(A, S)$ returns *true*, B is decrypted and counted. The complete tallying procedure runs in $\mathcal{O}(N + \beta n)$ time.

4.1.4. SKHS12: Enhanced Scheme by Spycher, Koenig, Haenni, Schlaepfer

In contrast to the former scheme, SKHS12 [107] allows to shift the computing power needed away from the tallying phase into the setup phase. Furthermore, there is no negative impact at the voter's side in terms of computational power nor in terms of coercion-resistance. This is achieved by introducing an indicator i which marks a credential σ (see Figure 4.3). During setup phase of the system, a list of tuples $(Enc(\sigma), Enc(i))$ containing an encrypted σ and an encrypted i is created. This list is a constant factor β times bigger than the voter roll, hence *beta* defines the size of the anonymity-set. Just before voters can register, a new permuted list containing all indicators i is produced and publicly announced as $\mathcal{UNL} < i >$ (see Figure 4.3).

In addition to σ, each voter receives during registration the corresponding i via an untappable channel. If coerced directly after stepping out of the untappable channel, the voter can lie about i by simply handing over a different but valid $i' \in \mathcal{UNL} < i >$.

In order to cast a valid vote, the ballot is adjoined with an encryption of σ and the corresponding indicator in plaintext. After duplicate elimination, just before the mixnet, the indicator i on each ballot is replaced by the corresponding $Enc_y(\sigma)$ from the $\mathcal{UNL} < E(\sigma, i) >$ and forms the special set $\mathcal{B}_2^*$. This way, the filter responsible for fake vote elimination, starts by verifying if the placed encrypted σ is the same as the credential encrypted in A on the cast ballot. This is done by executing a single PET per vote cast. Now the same filter excludes valid credentials not being part of the electorate. These unauthorized ballots can be removed as in Smith's and Weber's schemes, as the described attack does not hold in this case. This last removal is required, to universally verify that only ballots from the actual electorate are taken into consideration, and thus to provably maintain democracy.

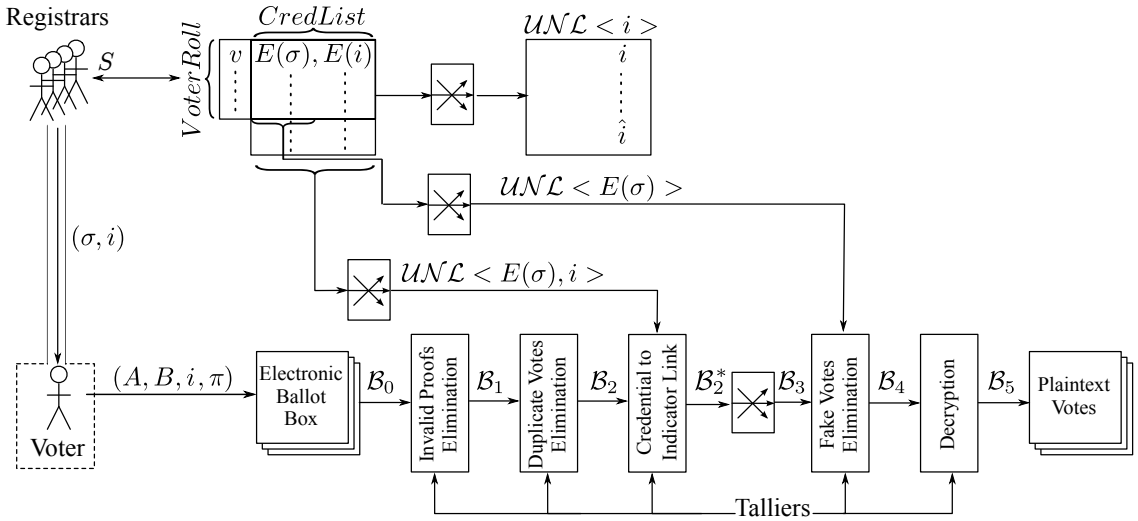


Figure 4.3.: Overview of the Enhanced Scheme: With the help of the enhancements, all filters work in linear time in respect of the cast votes.

This way, a voter under coercion can always deceive the coercer by presenting a different (existent) indicator from $\mathcal{UNL} < i >$ and a bogus credential. This enables the voter to anonymously cast the true vote in a moment of privacy even if the coercer constantly watches the public bulletin board. The complete tallying procedure runs in $\mathcal{O}(N)$ time.

4.2. Analysis of Privacy and Coercion-Resistance

In this section, privacy and coercion-resistance is analyzed with respect to the approaches presented in this chapter. The adversary model described in Section 3.3 is used, where it is assumed that a polynomial bounded adversary may corrupt a minority of registrars, but in a way that the set of corrupted registrars is known to the voter. The adversary may also corrupt a minority of talliers and arbitrarily many voters in a static, active manner. By corrupting a tallier, the adversary learns the corresponding share of the private key x and all secret randomizations. By requiring an untappable channel during registration and an anonymous channel during vote casting, it is assumed that the adversary learns nothing about the voter's private communications over these channels. This implies that during vote casting, every voter has access to the anonymous channel for silently casting at least one vote. Finally, with respect to the pressure exercised on voters under coercion, it is assumed that the adversary has limited resources in terms of time or money.

By applying this adversary model to the different schemes discussed within this chapter, their compliance with the notions of privacy and coercion-resistance as introduced in Subsection 3.3.1 will be analyzed. Arguments concerning the registration phase are excluded from the analysis, as they can be adopted from the original paper.

Coercion-Resistance in JCJ05-Based Schemes: What makes the original JCJ05-scheme coercion-resistant is the fact that any ballot sent to the electronic ballot box is accepted, if it is well-formed and complies with the scheme. This enables the voter to deceive the adversary with a randomly chosen fake credential (see Subsection 3.3.1), for example by using it for casting a vote that complies with the demands or by handing it over to the adversary in a simulation attack. Votes accompanied with such fake credentials are discarded during the tallying phase, but the two mix-nets involved in the tallying phase guarantee that no voter can prove to a third party whether a particular ballot has been discarded before tallying or not. This property finally prevents voters from selling their votes or from being coerced. Note that this conclusion holds for all three types of coercive attacks considered in Section 3.3.

Two of the linear-time schemes presented within this chapter offer an adjustable security parameter β to trade off coercion-resistance against efficient tallying, where β is the size of some anonymity set. To quantify coercion-resistance as a function of β, a game-theoretic definition given in [73] has been considered and extended, where the level of coercion-resistance a protocol provides is defined in terms of the *adversarial uncertainty*, i.e., the probability $\delta \in [0, 1]$ that the adversary is able to distinguish whether a coerced voter is following the instructions or running a counter-strategy. The quality of the counter-strategy directly depends on β, hence the possibility to successfully hide within the anonymity set. The attempt fails if multiple voters claim to 'be' a specific element within the anonymity set. Hence, the bigger the set, the lower the probability of a failing the attempt to convince the adversary of being 'someone else'.[1]

In all schemes offering a security parameter β (Sections 4.1.2, 4.1.3), maximal coercion-resistance of degree $\delta = 0$ is achieved by selecting $\beta \geq n$, but then the tallying procedure falls back to a quadratic running time. At the other extreme, selecting $\beta = 1$ (respectively $\beta = 0$, depending on the scheme) implies efficient linear-time tallying, but then coercion-resistance is ruled out entirely as δ rises towards 1. As tallying remains efficient as long as β remains constant, a reasonable level of coercion-resistance $0 < \delta \ll 1$ can be achieved by selecting β appropriately In the paper [107] Oliver Spycher introduced a discussion on the relationship between β and δ for those schemes resulting in the diagram in Figure 4.4.

[1] In all schemes, coercion-resistance is only reduced with respect to forced vote abstention.

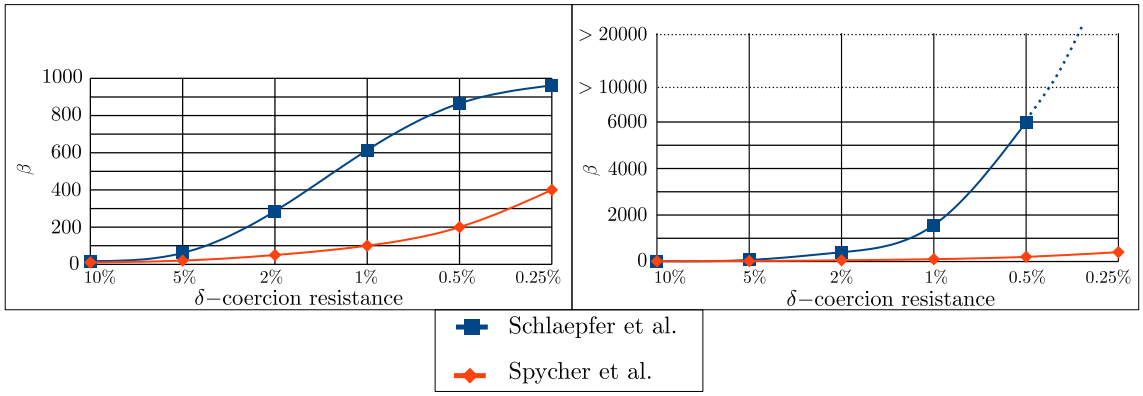


Figure 4.4.: The two drawings show the order of the β−set required in order to achieve a certain δ−coercion resistance. At the left-hand side, 1000 voters were involved whereas at the right-hand side, 100000 voters were involved.

4.3. Related Work

During the past four years, other groups have contributed on coercion-resistant E2E remote e-voting as well. Several works focussing on the linearization of JCJ05 have been presented by Araújo et al.. The group of Hengartner at the university of Waterloo (Canada), where Jeremy Clark and Aleksander Essex worked as Ph.D. students also provided several different approaches for the linearization of the JCJ05 tallying process.

Civitas The only true implementation of JCJ05 has been made by Clarkson et al. [28] made available in 2007. It splits the public board into smaller 'mail boxes' so the quadratic processing time for duplicate elimination and eligibility tests can be reduced substantially. However, this implementation remained on the academic level and has never been used for a true voting event.

AFT In 2007 Araújo et al. [3, 4] presented a new protocol with the same properties for coercion-resistance as provided by JCJ05 but working in linear time. However, their protocol allows the undetectable creation of illegitimate but valid ballots by the corrupted authority. Furthermore it bears an inherent usability problem, as it does not allow the revocation of a single member of the electorate, without urging the remaining members into another registration phase. In particular, in order to exclude a parting member of the electorate from further participation, each remaining member of the electorate must re-register on a new instance of the AFT-voting system.

Selections In 2011 Clark et al. presented a protocol called Selections [26]. Their approach is similar to the SKHS shemes, where the voter knows the position of the published encrypted credential (In selections this table of published encrypted credentials is called roster). But in contrast to the SKHS schemes, Selections uses ZKP in order to indicate the correct position on the roster, whereas SKHS schemes are using encryptions. This way, Selections also provides linear time on the tallying process, however requires more computing power on the client side.

Selections also introduces a new idea for the credential-handling on the client side by applying their findings on *panic passwords* [27].

COBRA Furthermore, they also have given birth to new ideas by finding new ways to speed up the ballot authorization process demonstrated in a protocol called COBRA [41]. This protocol however, remains purely theoretical and thus cannot be translated into a running system. Though, it brings in fresh ideas to the topic.

Trivitas In 2011 Bursuc et al. [18] described a JCJ05 protocol extension, allowing to distinguish between real credentials and trial credentials. This way, the protocol allows an individual audit during the voting event. An important aspect that establishes trust. Furthermore, the described protocol allows to decrypt all the data such that individual verifiability becomes easier for the human voter.

However, all approaches described so far do not provide a solution to the board-flooding and they do not provide a solution to unintended errors made by the human voter while processing the password.

Chapter 5

Rendering JCJ05-Schemes Board Flooding Resistant

The fact that no system is known to be in use implementing a linear derivative of JCJ05-scheme, indicates that there must be at least one other unsolved problem at the protocol level: Despite the fact that all these schemes are designed within the secure multi-party computation setup, they lack efficiency. This is due to the inherent possibility for all given schemes to post an unlimited amount of invalid (bogus) votes, and therefore to flood the public board on 'protocol-level' with many invalid votes (such that N becomes orders of magnitude larger than the size of the electorate n). This ultimately disables the tallying process and eventually violates democracy as the volition of the electorate cannot be reflected in acceptable time. The following sections state pioneering work aimed to remove this impracticability, which has been an open problem since the first days of JCJ05.

5.1. Board Flooding Resistant Schemes

The following section combines two publications. First a generic linear-time enhancement is described, suitable for any of the JCJ05-based schemes introduced previously to become resistant against board flooding attacks. This enhancement guarantees a hard upper limit for N, thus ensuring efficient tallying [56]. Then the special approach of [69] is described, integrating a similar mechanism into the original JCJ05-scheme.

The mathematics used within this section is used exemplary and targets an ElGamal-setup for the encryption-scheme. However, it can be adapted for any other probabilistic public-key encryption-scheme with homomorphic properties: Let the plaintext $\mathcal{M}$ states a multiplicative, modular group $\mathcal{G}_q$ where the discrete logarithm

is supposed to be hard, and the decisional Diffie-Hellman assumption holds. G_q contains the quadratic residues of $\mathbb{Z}_p^*$ where $\mathbb{Z}_p^*$ defines a muliplicativ modular group with prime-modulus $p = 2q + 1$ and q is prime. A generator g represents a primitive root of G_q.

5.1.1. Scheme by Haenni, Koenig: Generic Approach

The main idea of the following approach [56] is to protect the electronic ballot box against application-level flooding attacks. For this, it needs to be equipped with a stronger filter for what is an acceptable ballot. By introducing *posting tickets*, which are distributed by the registrars to the voters during the registration phase, ballots not accompanied by a valid posting ticket can be filtered out right from the beginning. To realize this filter, a combination of an exponentiation mix-net (Section 2.4) and an identification scheme is used. This idea is similar to the anonymous authentication technique proposed in 2010 by Spycher et al. [106] and 2011 by Haenni et al. [55]. During the mixing, a random exponent α is applied to a given list of public keys. If x and $y = g^x$ form an input key pair, then x and $\hat{y} = y^\alpha = \hat{g}^x$ form an output key pair with respect to a fresh generator $\hat{g} = g^\alpha$ ($\hat{g}$ is published along with the mix-net data without revealing α). In the generic approach, pairs (x, y) are used as private and public posting tickets. By applying an identification protocol, such as the Schnorr protocol [100] to prove knowledge of x with respect to $\hat{y}$ and $\hat{g}$, the voter is authenticated anonymously as a holder of a valid posting ticket. This allows the electronic ballot box to reject ballots with invalid proofs.

In the following, the required generic enhancement of the registration and vote casting phases (the tallying phase remains untouched) is introduced. An overview of the enhanced scheme is given in Figure 5.1.

Registration: In addition to the registration requirements of the JCJ05-based scheme in use, the registrars jointly establish a set of private posting tickets $X = \{x_i \in \mathbb{Z}_q : 1 \leq i \leq d\}$ and the set $Y = \{y_i \in G_q : 1 \leq i \leq d\}$ of corresponding public posting tickets $y_i = g^{x_i} \mod p$, where d might be different for every voter (see Subsection 5.2.1)[1]. X is delivered to the voter via an untappable channel and Y is added to the set $\mathcal{Y}_0$ of all public posting tickets for all voters, which resides on the public bulletin board. At the end of the registration phase, the complete set $\mathcal{Y}_0$ is digitally signed by the registrars. The total number of issued posting tickets, $D = |\mathcal{Y}_0|$, represents the maximum amount of ballots allowed on the electronic ballot

[1] Please note that all calculations are made using residue groups modulo p. However, the mathematics may be adapted to ones needs

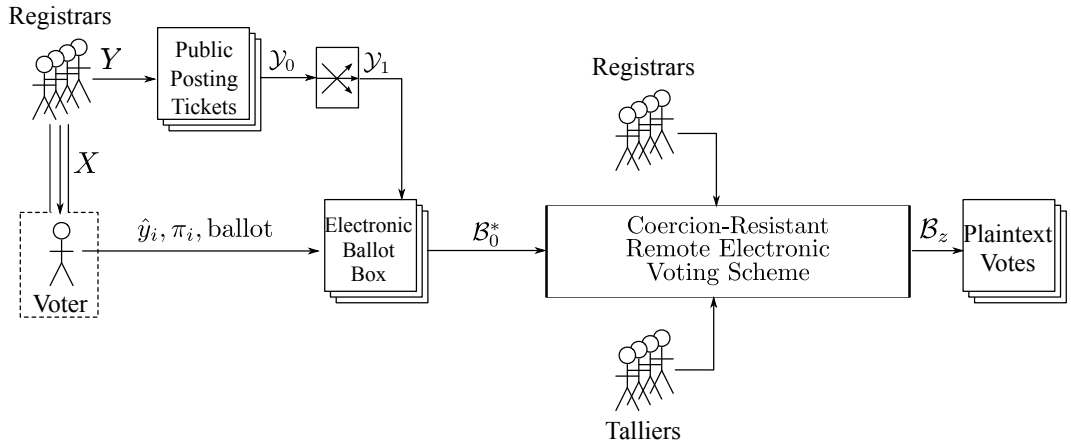


Figure 5.1.: Overview of the generic protocol enhancement: the ballot sent to the electronic ballot box is accompanied by an identification proof for an anonymized public posting ticket. Ballots with invalid proofs are rejected by the electronic ballot box.

box. At the end of the registration phase, $\mathcal{Y}_0$ is mixed in an exponentiation mix-net and the fresh generator $\hat{g}$ is published. $\mathcal{Y}_1$ denotes the output of the mix-net.

Vote Casting: To cast a vote using a private posting ticket x_i, the ballot of the JCJ05-based scheme in use is enhanced by $\hat{y}_i = \hat{g}^{x_i}$ and a proof $\pi_i = ZKP\{(x_i) : \hat{y}_i = \hat{g}^{x_i}\}$ of knowing x_i. The resulting enhanced ballot is accepted by the electronic ballot box, if the verification of the enhanced proof succeeds and if $\hat{y}_i \in \mathcal{Y}_1 \setminus \mathcal{Z}$, where $\mathcal{Z}$ denotes the set of public posting tickets already appearing in the electronic ballot box. Finally, all $N \leq D$ accepted ballots $\mathcal{B}_0$ are passed without the enhancement to the tallying phase of the JCJ05-based scheme in use.

By applying the generic approach introduced above to any of the existing JCJ05-based schemes, the maximum number of ballots in the tallying phase is restricted to D, which implies that N results in $\mathcal{O}(n)$ if D is constant.

Tallying: The generic approach does not have any influence on the tallying phase except that it guarantees an upper limit of votes having to be processed by the underlying scheme.

Revocation: In case of revocation due to the elimination of a member of the electorate, simply removing all according posting credentials would leak information

about the amount of posting tickets the member was in possession of. Hence, the posting tickets have to remain in the system until there is at least another change in the electorate (insertion or elimination of a member). So, maintaining democracy after revocation has to be handled by the underlying scheme.

5.1.2. Scheme by Koenig, Haenni, Fischli: Integrated Approach

Applying the generic approach to the original JCJ05-scheme, the tallying phase still requires $\mathcal{O}(n^2)$ calculations. This is due to the expensive PETs required during the fake credential removal. In the scheme addressing the board flooding problem for the first time [69], the idea of the generic approach and the original JCJ05-scheme are integrated more tightly, resulting in a linear-time tallying phase. In this *integrated approach*, a set of so-called *dummy credentials* is distributed to the voters during the registration phase (together with the proper secret credential). Ballots not containing a proper or a dummy credential can then be rejected by the electronic ballot box during vote casting. Dummy credentials are used to produce fake votes. In other words, a voter can post several ballots to the electronic ballot box, but only the one containing the proper credential will make it to the final tally. This way, the credentials themselves bear the additional property provided by the posting tickets in the generic approach. An overview of this scheme is given in Figure 5.2.

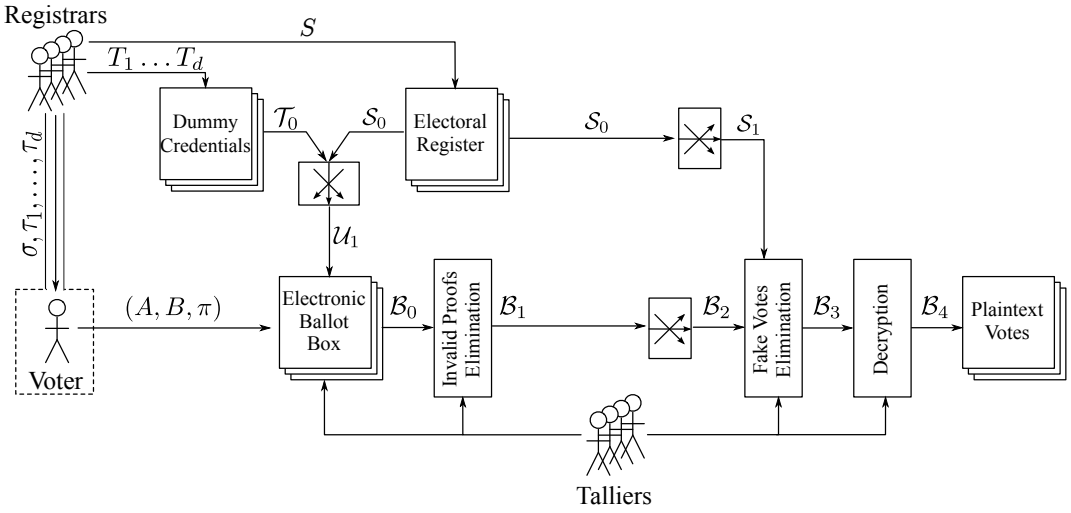


Figure 5.2.: Overview of the integrated approach: ballots not containing a proper credential or containing a dummy credential as well as duplicate votes are rejected by the electronic ballot box with the help of the talliers.

Registration: Let $\{\tau_i \in G_q : 1 \leq i \leq d\}$ be the set of dummy credentials for a given voter. They are generated jointly by the registrars together with the secret credential σ and distributed over an untappable channel. As in the generic approach, the number of dummy credentials d might be different for every voter. Corresponding encryptions $T_i = \mathrm{Enc}_y(\tau_i, r_{T_i})$ are published on the public bulletin board. The set of all dummy credentials published on the board is denoted by $\mathcal{T}_0$ and its size by $D = |\mathcal{T}_0|$. If the set of proper credentials is denoted by $\mathcal{S}_0$, as in the description of the original JCJ05-scheme (Subsection 3.3.1), then $\mathcal{U}_0 = \mathcal{S}_0 \cup \mathcal{T}_0$ represents the complete set of available credentials. The size of this set, $U = |\mathcal{U}_0| = n + D$, represents the maximum number of ballots in the electronic ballot box. By applying a verifiable re-encryption mix-net to $\mathcal{U}_0$ at the end of the registration phase, a new set $\mathcal{U}_1$ is obtained.

Vote Casting: The ballot is constructed as in the original JCJ05-scheme. To fake a vote, one that will appear on the public bulletin board but not in the final tally, a dummy credential τ_i is taken in place of σ. In either case, the resulting ballot (A, B, π) is sent to the electronic ballot box $\mathcal{B}_0$. It is accepted if $PET(A, U) = \mathit{true}$ holds for some $U \in \mathcal{U}_1$ and if no other such match for U has been found before. To perform these tests in linear time, Smith's and Weber's technique can safely be applied here, because the votes have not yet been mixed (and so the voter-designated link between the voter and the corresponding vote is still prominent). Note that this step requires the help of the talliers already during the vote casting phase, which marks a disadvantage compared to the generic approach. Another consequence is the fact that the electronic ballot box does not accept and therefore will not hold any duplicate votes.

Tallying: Four consecutive steps are necessary to single out the valid votes from the list of ballots in the electronic ballot box (Figure 5.2):

1. The proofs π are verified for all ballots $(A, B, \pi) \in \mathcal{B}_0$. Ballots for which the proof does not hold are excluded from further processing. The remaining reduced ballots (A, B) form a new set $\mathcal{B}_1$.

2. The sets $\mathcal{B}_1$ and $\mathcal{S}_0$ are mixed in two separate verifiable re-encryption mix-nets. These mix-nets produce two new sets $\mathcal{B}_2$ and $\mathcal{S}_1$.

3. Ballots containing a dummy credential are excluded from further processing. This is the case for a ballot $(A, B) \in \mathcal{B}_2$, if $PET(A, S) = \mathit{false}$ holds for every

$S \in \mathcal{S}_1$. Again Smith's and Weber's technique can safely be applied to perform this step in linear time.[2] The remaining encrypted votes B form a new set $\mathcal{B}_3$.

4. The encrypted votes $B \in \mathcal{B}_3$ are jointly decrypted. This yields a new set $\mathcal{B}_4$, which contains the plaintext votes ready to be counted.

Let $N = |\mathcal{B}_0|$ denote the number of ballots in the initial electronic ballot box. Clearly, the tallying procedure runs in $\mathcal{O}(N)$ time, where $n + |\mathcal{T}_0|$ defines an upper limit for N. If the average number of dummy credentials per voter is constant, this implies that finally $\mathcal{O}(n)$ is obtained for the tallying phase.

Revocation: In case of revocation, the integrated approach only revokes the corresponding σ. However, it is not allowed to simply remove its encryption from $\mathcal{S}_0$, but a re-encrypted version of the σ in focus has to be included in $\mathcal{T}_0$. This way, the credential remains valid but can only be used to fake a vote.

5.2. Analysis of Privacy and Coercion-Resistance

In the following, the same security analyses regarding privacy and coercion-resistance is done as in chapter 4. The very same adversary model will be used as in Section 4.2, this time, with respect to the parts providing practical efficiency. So, the following lines state a direct copy from Section 4.2 for convenience reasons to the reader:
The adversary model described in Section 3.3 is used, where it is assumed that the polynomial bounded adversary may corrupt a minority of registrars, but in a way that the set of corrupted registrars is known to the voter. The adversary may also corrupt a minority of talliers and arbitrarily many voters in a static, active manner. By corrupting a tallier, the adversary learns the corresponding share of the private key x and all secret randomizations. By requiring an untappable channel during registration and an anonymous channel during vote casting, it is assumed that the adversary learns nothing about the voter's private communications over these channels. This implies that during vote casting, every voter has access to the anonymous channel for silently casting at least one vote. Finally, with respect to the pressure exercised on voters under coercion, it is assumed that the adversary has limited resources in terms of time or money.

By applying this adversary model to the different schemes discussed in this chapter, their compliance with the notions of privacy and coercion-resistance as introduced in Subsection 3.3.1 will be analyzed. Arguments concerning the

[2] The attack described in [89] does not apply here, because voters cannot freely choose related plaintext credentials.

registration phase are excluded from the analysis, as they can be adopted from the original paper.

5.2.1. Coercion-Resistance in the Generic Approach

In the generic approach, the voter can submit a limited amount of ballots to the electronic ballot box, depending on the number of posting tickets received during registration. As a consequence, if the voter can be forced to use all posting tickets with invalid votes, then the possibility of submitting a final valid vote is denied. This restriction enables forced-abstention attacks. To run such an attack, the adversary may offer a certain amount of money for each posting ticket received from the voter or spent by the voter on an invalid vote. From an economic perspective, this means that restricting the number of posting tickets makes them valuable. If $\mu = {}^{D}/_{n}$ denotes the average number of posting tickets per voter, where $D = |\mathcal{Y}_0|$ is the total number of posting tickets and n the number of voters, then increasing μ decreases the value of each issued posting ticket, and vice versa. In contrast to the linear schemes presented before, this scheme does not depend on an anonymity set β, but this time μ plays the role of an additional security parameter, influencing δ. In the following, answers are given to some questions on how to choose μ and on how to generally deal with posting tickets.

How many posting tickets are needed?

To answer this question, suppose first that each voter receives the same number $d \geq 1$ of posting tickets. Then the adversary can simply force the voter to release all d posting tickets and check their validity by observing if corresponding ballots are accepted by the electronic ballot box. However, for the same reasons as in the original JCJ05-scheme, releasing the secret voting credential σ cannot be enforced, i.e., the voter is still protected from being coerced, except for forced vote abstention.

As a counter-measure against forced-abstention attacks, the registrars have to issue a random number of posting tickets to each voter. Suppose that a given voter receives $d \in \{1, \ldots, d_{max}\}$ posting tickets, where d_{max} denotes a fixed upper limit for all voters. If the scheme guarantees that d is not known to the adversary, then the voter can lie about it, for example by releasing only $d - 1$ posting tickets to the adversary. Obviously, this argument works for every voter possessing $d > 1$ posting tickets and thus completely rules out coercion in those cases. Unfortunately, this is not true for voters possessing exactly $d = 1$ posting ticket. Under coercion, such (unfortunate) voters could only give away a single posting ticket, which means that they would be unable to cast the final vote. In other words, $d = 1$ makes voters prone to coercion by forced vote abstention. Note that this problem does not disappear by increasing the lower limit of d to some value $d_{min} < d_{max}$ or by decreasing it to 0.

Another problem exists for voters in possession of $d = d_{max}$ posting tickets. They can prove the release of all posting tickets to a potential adversary paying for vote abstention.

As an answer to the above problems, it is suggested that d is selected according to some non-uniform probability distribution over $\mathbb{N}_1 = \{1, \ldots, \infty\}$. The most natural choice is a chi distribution or one of its derivatives. Within the reviewed paper the calculations were made using the more general normal distribution $\mathcal{N}(\mu, \sigma^2)$ with reasonable mean $\mu > 0$ and variance $\sigma^2 > 0$. Since normal distributions are defined by continuous probability density functions $f : \mathbb{R} \to [0, 1]$ or corresponding cumulative density functions $F : \mathbb{R} \to [0, 1]$, they need to be applied in some discretized manner over $\mathbb{N}_1$. For this, let

$$f'(x) = c^{-1} \cdot f(x) \cdot H(x)$$

be the truncated distribution over $\mathbb{R}^+$, where $H(X)$ is the heaviside step function

$$H(n) = \begin{cases} 0, & n \leq 0, \\ 1, & n \geq 0, \end{cases}$$

and

$$c = \int_0^\infty f(x)dx = 1 - F(0)$$

the normalization constant. Then one can discretize f' into $f^* : \mathbb{N}_1 \to [0, 1]$ by

$$f^*(x) = \int_{x-1}^{x} f'(x)dx = \frac{F(x) - F(x-1)}{1 - F(0)} = \frac{\Phi(\frac{x-\mu}{\sigma}) - \Phi(\frac{x-\mu-1}{\sigma})}{\Phi(\frac{\mu}{\sigma})},$$

where Φ denotes the cumulative distribution function of the standard normal distribution $\mathcal{N}(0, 1)$, and interpret $f^*(d)$ as the probability of obtaining exactly d posting tickets from the registrars.

How does a given distribution affect coercion-resistance?

The advantage of using a distribution f^* with no upper limit is that no voter can prove the release of all posting tickets. The challenge then is to adjust the available parameters μ and σ^2 such that only very few voters receive the minimal number of posting tickets, but without ruling it out entirely.

Assuming that d is unknown to the adversary, the voter's best counter-strategy against forced vote abstention is to release only $d - 1$ posting tickets, thus saving

one for the final vote. As mentioned above, this strategy does not work for voters possessing a single posting ticket only. Therefore, the adversarial uncertainty δ depends on the voter under coercion. If d_i denotes the number of posting tickets of voter i, then

$$\delta_i = \begin{cases} 1, & \text{if } d_i = 1, \\ 0, & \text{if } d_i > 1. \end{cases}$$

denotes the adversarial uncertainty with respect to a single voter, and

$$\delta = \frac{1}{n} \sum_{i=1}^{n} \delta_i = f^*(1) = 1 - \frac{\Phi(\frac{\mu-1}{\sigma})}{\Phi(\frac{\mu}{\sigma})}$$

is the *average adversarial uncertainty* over the entire electorate. Some example values for δ are shown in Table 5.1. To minimize δ, one can either choose a large mean or a small variance. Note that making the mean too large will harm the efficiency of the tallying phase, and making the variance too small will lead to almost the same number of posting tickets for every voter. In the extreme case when σ^2 tends towards 0, which implies that δ tends towards 0, it even seems that coercion is completely ruled out, but then exactly the same number of posting tickets is issued to all voters, which corresponds to the first unpleasant scenario discussed in our analysis. The problem is that δ reflects coercibility with respect to a single voter only, which is not a characteristic measure for statistical attacks on groups of voters. To deal with such attacks, it seems that a large variance is desirable, but here no measure dealing with statistical attacks is introduced. Therefore, forced vote abstention cannot entirely be ruled out by minimizing δ; but a careful selection of the security parameters μ and σ^2 can make it unbearable for the vast majority of voters (without spoiling the tallying procedure).

	$\sigma^2 = 1$	$\sigma^2 = 2$	$\sigma^2 = 3$	$\sigma^2 = 4$	$\sigma^2 = 5$	$\sigma^2 = 10$
$\mu = 1$	0.4057	0.3423	0.3038	0.2769	0.2567	0.1988
$\mu = 3$	0.0214	0.0628	0.0861	0.0984	0.1051	0.1112
$\mu = 5$	$3.1 \cdot 10^{-5}$	0.0021	0.0085	0.0166	0.0245	0.0488
$\mu = 10$	≈ 0	$9.7 \cdot 10^{-11}$	$9.8 \cdot 10^{-8}$	$3.1 \cdot 10^{-6}$	$2.5 \cdot 10^{-5}$	0.0014
$\mu = 20$	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	$8.1 \cdot 10^{-10}$

Table 5.1.: The adversarial uncertainty δ for some example security parameters μ and σ^2.

How do registrars generate random numbers of posting tickets?

The naïve approach for the registrars to generate a random number of posting tickets for a given voter is to jointly apply F to determine d. The problem of this simple approach is that then d is not a secret of the voter alone, i.e., the voter cannot lie about it to an adversary colluding with one of the registrars. As a solution to this problem, it is suggested to split up the group of registrars into r sub-groups. Each of these sub-groups is then responsible for secretly generating an average of d/r posting tickets, but without informing the other groups about the exact number. To do so, the normal distribution $\mathcal{N}(\mu, \sigma^2)$ is decomposed into a sum of r normal distributions $\mathcal{N}(\mu/r, \sigma^2/r^2)$. Each sub-group generates its own subset of posting tickets and communicates them separately to the voter over the untappable channel. Coercion resistance is maintained, if at least one sub-group remains honest.

How should the public bulletin board store the encrypted posting tickets?

In the original JCJ05-scheme, the list of encrypted credentials $\mathcal{S}_0$ is published in the electoral register together with the plaintext identities of the voters. This list resides on the public bulletin board and can be inspected and verified by everybody. Doing the same with the posting tickets y_i by linking $Y = \{y_1, \ldots, y_d\}$ publicly with the voter's identity allows the adversary to derive the secret number d from Y. As a simple solution to this problem, Y is published anonymously in $\mathcal{Y}_0$ without any links to the voters. Since $\mathcal{Y}_0$ does not serve as an electoral register, it does not necessarily need to be treated in exactly the same way as $\mathcal{S}_0$.

How can the voter hide the number of posting tickets?

During registration, the voter receives d posting tickets over an untappable channel. To omit vote-abstention attacks, the voter has to be able to hide d from the attacker. Therefore, the voter is required to manage each posting ticket independently, so that disclosing one posting ticket does not infer the existence of another. In practice, especially if d is large, it is difficult for the voter to realize such a management in a usable way. As a possible approach, a cryptographic component is suggested similar to an encrypted password vault, but with the additional property that the extraction of a single secret does not disclose any information about the remaining secrets and that the exact number of secrets always remains hidden.

An approach for a system with these properties based on polynomial interpolation is presented in Part II.

Is the chosen adversary model appropriate?

In contrast to the original JCJ05-scheme, the generic approach works with probabilities. This should also be reflected in the adversary model. Hence the coercion of a single voter cannot be extrapolated to a group of voters, as the law of large numbers [85], stating a fundamental theorem of probability, gains weight. If the adversary manages to figure out the distribution properties, the coercer urges the group of voters to bring forth a number of posting tickets which is slightly above the average, but still within the variance. This way some voters will have to hand out all posting tickets. A property not applicable in the original JCJ05-scheme. However, as the adversary cannot control the complete electorate (by definition), the voters under coercion might ask non coerced members of the electorate to help them out with some of their posting tickets in order to deceive the coercer. Even though this seems possible in theory, this 'trick' sounds rather complicated—even to me. On the other hand, one must confess, that the probability of coercing a larger group of the electorate without being noticed universally, is decreasing substantially. So the question remains: Is there an appropriate adversary model able to somewhat reflect reality?

5.2.2. Coercion-Resistance in the Integrated Approach

By issuing dummy credentials instead of posting tickets, the integrated approach limits the number of ballots the voter can submit to the electronic ballot box in a similar way as in the generic approach. If $\mu = {}^{D}/_{n}$ denotes the average number of dummy credentials per voter and $D = |\mathcal{T}_0|$ the total number of dummy credentials, then μ plays the role of a security parameter influencing δ as in the generic approach. This raises the same questions about how to choose μ and about how to generally deal with dummy credentials. All conclusions from the previous subsection can be adopted, except those concerning the impact of a successful coercion attack.

Considering the attack from the generic approach, where voters possessing a single posting ticket can be forced to abstain from voting. In the integrated approach, the same attack enables the adversary to gain possession of both the single dummy credential and the secret credential, simply by forcing the voter to release two credentials. Their validity can then be checked by sending respective ballots to the electronic ballot box and by observing if they are accepted. Therefore, voters in possession of only $d = 1$ dummy credential are exposed to all four types of coercive attacks. However, using a normal distribution with appropriate parameters for picking d limits the scalability of this attack in the same way as in the generic approach.

5.3. Summary

The quadratic tallying phase of the JCJ05-scheme does not allow an implementation usable for large scale elections. However, as JCJ05 is the only E2E-verifiable[3] coercion-resistant remote e-voting scheme known, the linearization of the tallying phase and the setting of a hard upper bound for the tallying phase is a must. Even if coercion-resistance cannot be maintained on the same level as in the original scheme, the contributions described above allow a parameterizable level of coercion-resistance in respect of tallying time using reasonable parameters. Unfortunately, this comes with a new level of scheme complexity sensitive to even the slightest alteration during implementation of a corresponding protocol. Finally, the introduction of probability within the schemes asks for a modified attack and defense model, as the original model given by Juels et al [63] is not able to cover this aspect.

[3]This statement is only true if the voter is considered to be perfect in respect to the processing of high entropy values. It fails if a more realistic model for the human voter is in use as is demonstrated in Part II

Part II.

Managing High Entropy Credentials in the Context of Coercion

Chapter 6

A Multi-Encryption Scheme and its Implementation

In the case of the protocols described in Chapter 5, each voter needs to protect multiple high-entropy credentials in a way that the voter can handle each of the credentials but at the same time is not able to signal the total amount of managed credentials to anyone else. As the capabilities of the human brain are quite limited when it comes to memorizing high entropy credentials they must be managed by a *credential management scheme*, i.e. some construct serving as the interface between the needs of cryptography and the human brain. But how can these credentials be managed safely when facing the adversary at some moment in time?

6.1. Where to Store the Credential

There are two main ideas on how to protect the right to vote:

Physical Device: A physical, tamper-proof device as described by Schweissgut [102], can internally keep the high entropy credentials, while the voter accesses them using a somewhat low entropy password. If this device allows only to enter a restricted amount of passwords per time unit, or even stops working at all when challenged too often, the adversary cannot attack the device with a brute-force attack. Furthermore the physical nature of such a device spoils any attempt for a distributed attack. However, such a device allows the adversary to provoke a vote abstention by simply forcing the voter to hand over the device with the effect that the voter is no longer able to vote. This kind of attack is possible for any tamper-proof device a system relies on, as it is not a simple process to get a new device for the voter as well as for the underlying system. As such a device cannot be read out and as it

should be the only device that knows the voting credential, the production of a clone is not possible by design. This way, the voter would have to go through the whole setup procedure again in order to regain the right to vote by contacting the authorities, over the untappable channel. The authorities would have to revoke the voter's credential and then would have to create a new one. This is highly inefficient, very cost intensive in terms of time and money and error-prone.

Cryptographic Component: The alternative to the physical device is a cryptographic component. Of course, the voter will not interact with the cryptographic component directly, but can use a simpler physical device able to read in the cryptographic ciphertext covering the credential. This way, the voter can again interact with a physical device and the credential can be revealed by entering a password. As the cryptographic ciphertext is available publicly (Kerckhoffs's principles), the adversary cannot provoke a vote abstention attack any more. If the physical device managing the ciphertext is forced from the voter, the voter simply loads another device with the ciphertext. However, the attacker is now able to start a distributed brute-force attack on the public ciphertext, thus the system has to be resistant to such attacks.

The clear shortcomings shown above of a tamper-proof physical device urges the use of the cryptographic component. However, this component must show some special properties and has to withstand the described attack on the polynomially bounded pseudo-offline adversary (see Section 2.3).

Within this chapter, a novel cryptographic component, namely the *multi-encryption scheme* [67, 70] is introduced, providing the required properties. It is very surprising, that this component has not been described in the cryptographic literature so far. The properties and definitions that follow come purely from the needs encountered while trying to render board flooding resistant JCJ05-schemes usable for the voter and thus might be incomplete for a general description of this component.

6.2. Properties

In contrast to a classical symmetric cryptosystem where the focus lies on the encryption and management of a single plaintext, a multi-encryption scheme allows to cryptographically manage multiple plaintexts independently. However, a multi-encryption scheme cannot be constructed by simply concatenating n ciphertexts created by a classical cryptosystem using n different keys for n plaintexts. This way

the decryption of a single plaintext k_i would require to either know the position within the list of ciphertexts, or by the decryption of all ciphertexts. Note that a multi-encryption scheme is also different from a single symmetric cryptosystem, which is used to encrypt a list of plaintexts $M \in \mathcal{M}^n$ with a single master key $k \in \mathcal{K}$. This is the functionality generally provided by a password vault system. Clearly, classical symmetric cryptosystems and password vault systems are both special cases of a multi-encryption scheme of order $n = 1$. A multi-encryption scheme thus has to provide the following properties:

Multiple Plaintexts: The scheme must be able to hold and efficiently manage an arbitrary amount of plaintexts.

Plaintext Hiding: The scheme must not reveal information about the plaintexts it manages.

Quantity Hiding: The scheme must not reveal information about the true amount of encrypted plaintexts it manages.

Independence of Keys: Knowledge of one or several keys of a given ciphertext must not reveal information about any remaining keys or plaintexts.

No Search: Applying a genuine key to a multi-encryption directly reveals the corresponding plaintext, without the need of further discrimination between the plaintexts. This very special feature states a key property of multi-encryption schemes and is in accordance to the works of Stajano [111] allowing humans to deal with high-entropy data serving as secret-keying material for some cryptographic system.

Indistinguishability of Keys: False keys must return plaintexts indistinguishable from actual plaintexts. This ensures that an adversary cannot distinguish false keys from genuine keys. This property must hold true even if the plaintext alphabet is limited.

Reusability of Keys: It must be possible to reuse the keys used within one ciphertext in other ciphertexts without any loss of security. Thus, it must be secure against differential cryptographic analyses.

Even though this kind of scheme seems to have quite a potential in its applications, this chapter will only cover the form required in order to manage the high entropy credentials of JCJ05 and derivatives in the context of coercion. After the formal definition of the multi-encryption scheme a concrete mathematical implementation will be given in order to prove its existence.

6.3. Definition

As briefly described in Section 2.4, there are two forms of ciphertexts, namely the deterministic and the randomized form. Reconsidering the definitions made in 2.4, let *plaintext space* $\mathcal{M}$ be the set of all possible plaintexts, and a *key space* $\mathcal{K}$ the set of all possible keys. The following passages up to Subsection 6.5.1 are excerpt of our paper [70]: Let $M = (m_1, \ldots, m_n), m_i \in \mathcal{M}$ be a list of $n \geq 1$ (not necessarily distinct) plaintexts and $K = (k_1, \ldots, k_n), k_i \in \mathcal{K}$ an equally long list of distinct keys. Furthermore, let $\mathcal{C}$ be the *ciphertext space*, the set of all possible ciphertexts.

The requirement of the system to be (t, ϵ)-secure ($\epsilon = 2^n$) also comprises the finding of k_i for a given ciphertext c and all according elements $(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_n)$ of K.

6.3.1. Deterministic Multi-Encryption Scheme

The two main components of a multi-encryption scheme are functions for encrypting and decrypting plaintexts. Formally, let

$$\mathsf{Enc} : M \mapsto_K c$$

denote the *multi-encryption function*, and

$$\mathsf{Dec} : c \mapsto_{k_i} m_i$$

the *decryption function*. As a notational convention, the functions are defined as, $c = \mathsf{Enc}_K(M) \in \mathcal{C}$ for encrypting a list of plaintexts $M \in \mathcal{M}^n$ with a list of distinct keys $K \in \mathcal{K}^{(n)}$ and $m_i = \mathsf{Dec}_{k_i}(c) \in \mathcal{M}$ for the corresponding decryption with key $k_i \in K$. Clearly, it is required that

$$\mathsf{Dec}_{k_i}(\mathsf{Enc}_K(M)) = m_i$$

holds for all $1 \leq i \leq n$, where n denotes the *size* of c.

To render a multi-encryption scheme usable for its intended purposes, it must possess the cryptographic properties of a traditional symmetric cryptosystem [81]. As a consequence, knowing c must not unveil any information about M (except possibly its length), or in technical terms, that $H(M|c)$, the conditional Shannon entropy of M given c, is equal to $H(M)$. More generally, consider a non-empty subset of indices $I \subseteq \{1, \ldots, n\}$ and corresponding sub-lists $M_I \subseteq S$ and $K_I \subseteq K$ of length $s = |I|$. Furthermore, let $I\text{-}\mathsf{Enc} : \mathcal{M}^s \times \mathcal{K}^{(s)} \longrightarrow \mathcal{C}$ be the *partial multi-encryption function* derived from Enc by choosing m_i and k_i arbitrarily for all $i \notin I$, and let $I\text{-}\mathsf{Dec} : \mathcal{C} \times \mathcal{K}^{(s)} \longrightarrow \mathcal{M}^m$ be the *extended decryption function* obtained from applying Dec for each $k_i \in K_I$ in the given order. This implies

$I\text{-}\mathsf{Dec}_{K_I}(I\text{-}\mathsf{Enc}_{K_I}(M_I)) = M_I$. Therefore, we require that I-Enc together with I-decrypt satisfies the cryptographic properties of a symmetric cryptosystem for all non-empty subsets $I \subseteq \{1,\ldots,n\}$. In particular, decrypting some plaintexts from c must not disclose any information about the other plaintexts in c, or in technical terms, that $H(M_I|c, M_J) = H(M_I)$ holds for all complementary subsets $I, J \subseteq \{1,\ldots,n\}$. As a consequence, $H(m_i|c) = H(m_i)$ and $H(m_i|c, m_j) = H(m_i)$ must hold individually for every $m_i \in M_I$ and $m_j \in M_J$.

Definition 6.3.1 *A* multi-encryption scheme *of order n,*

$$\Sigma[n] = (\mathcal{M}, \mathcal{K}, \mathcal{C}, \mathsf{Enc}, \mathsf{Dec}),$$

consists of a plaintext space $\mathcal{M}$, a key space $\mathcal{K}$, a ciphertext space $\mathcal{C}$, and two functions Enc *(with two arguments of arity n) and* Dec *with properties as introduced above.*

6.3.2. Randomized Multi-Encryption Scheme

A multi-encryption scheme according to the above definition is *deterministic*, i.e., encrypting a given list of plaintexts M with a given list of distinct keys K always results in the same ciphertext c. Some applications, however, may require a randomized multi-encryption function. Formally, let $\mathcal{R}$ be a *randomization space*, the set of all possible randomizations be R, and

$$\mathsf{rEnc} : M \mapsto_{K,R} c$$

be the resulting *randomized multi-encryption function* $rEnc_K(M, R)$ (the function Dec remains unchanged). As in the deterministic case, we expect

$$\mathsf{Dec}_{k_i}(\mathsf{rEnc}_K(M, R)) = m_i$$

to hold for all $1 \leq i \leq n$ and $R \in \mathcal{R}$. It is assumed that the same cryptographic properties hold with respect to corresponding functions I-rEnc and I-Dec. Finally, it is required that different randomizations distribute corresponding ciphertexts uniformly over $\mathcal{C}$, i.e.,

$$P(\mathsf{rEnc}_K(M, R) = \mathsf{rEnc}_K(M, R')) = \frac{1}{|\mathcal{C}|}$$

is the probability of getting the same ciphertext for two different randomizations $R \neq R'$.

Definition 6.3.2 *A* randomized multi-encryption scheme *of order* n*,*

$$\tilde{\Sigma}[n] = (\mathcal{M}, \mathcal{R}, \mathcal{K}, \mathcal{C}, \mathsf{rEnc}, \mathsf{Dec}),$$

consists of a plaintext space $\mathcal{M}$*, a randomization space* $\mathcal{R}$*, a key space* $\mathcal{K}$*, a ciphertext space* $\mathcal{C}$*, and two functions* rEnc *(with the first and the last argument of arity* n*) and* Dec *with properties as introduced above.*

6.4. Implementation

In this section, a concrete instance of a multi-encryption scheme is introduced. This specific implementation combines some standard cryptographic functions. It consists of any collision resistant [1] cryptographic hash function and any standard symmetric encryption scheme (e.g. AES). The resulting ciphertext consists of the coefficients of an interpolation polynomial used to provide the 'no search' property. The interpolation polynomial can be a Lagrange interpolation polynomial generated by an instance of the *Lagrange*-function.

The general idea of the approach corresponds to the Reed-Solomon coding scheme [92], which is based on polynomial interpolation over a finite field. To create a multi-encryption ciphertext, Enc will therefore create n symmetric-encryptions of the n plaintexts using the n keys; then it constructs a polynomial that contains a point for each of the n symmetric ciphertexts consisting of the tuple $(hash(k_n), symEnc(m_n, k_n))$. For decryption of a single plaintext from the ciphertext, Dec simply applies the polynomial on the corresponding hashed key and then decrypts the resulting $Dec(c, k_n) = symDec(c(hash(k_n)), k_n)$.

Please note, that there is no direct relationship between this implementation and the secret-sharing scheme by Shamir, except that both schemes are derivatives of Reed-Solomon Codes. In contrast to Shamir's scheme, where the polynomial has to be kept secret, here, the polynomial is part of the ciphertext (publicly known) but the sample positions must be kept secret.

6.4.1. Setup

Let p be a large prime number and $\mathbb{Z}_p$ the corresponding finite field of integers modulo p. The set of all possible polynomials over $\mathbb{Z}_p$ is denoted by $\mathbb{Z}_p[x]$ and serves as ciphertext space $\mathcal{C}$. $|\mathcal{C}|$ has to be at least as big as the ciphertext space consisting of all possible encryptions of the elements within the required plaintext space.

[1] In fact it only has to be collision resistant mod p per key-set K in use. Hence collision resistance is not required throughout multiple instances.

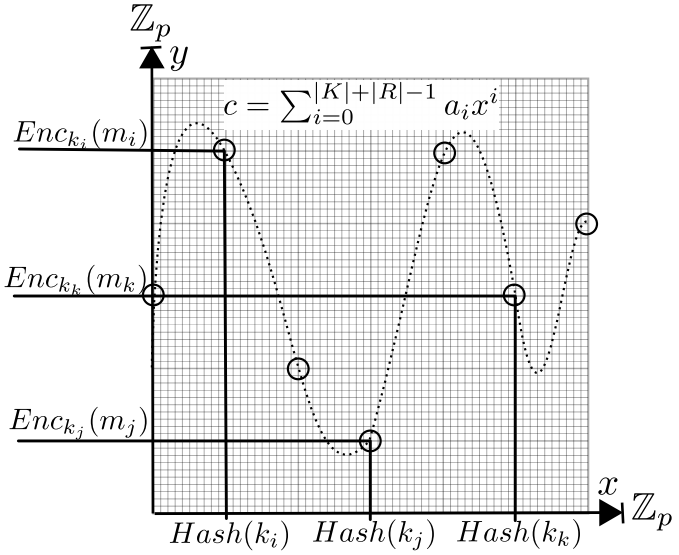


Figure 6.1.: Multi-Encryption scheme with 3 encrypted messages and 3 hashed keys placed within $\mathbb{Z}_p$. The resulting coefficients of the interpolation polynomial (indicated by the dotted line) form the ciphertext c. The circles indicate all possible tuples (k, m) of the given c, hence all possible messages $m \in \mathcal{M}$ reachable by any $k \in \mathcal{K}$.

6.4.2. Encryption

The following algorithms describe the basic encryption functions. The cryptographic hash-function $Hash(k_i)$ must provide a co-domain of order p in order to get a ditribution of k_i' uniformly over $\mathbb{Z}_p$.

Function $\mathsf{Enc}_K(M)$
Input: $M = (m_1, \ldots, m_n)$, $K = (k_1, \ldots, k_n)$
begin
  for $i \leftarrow 1$ **to** n **do**
    $k_i' \leftarrow Hash(k_i)$
    $m_i' \leftarrow symEnc_{k_i}(m_i)$
  $c(x) \leftarrow \mathsf{Lagrange}((k_1', m_1'), \ldots, (k_n', m_n'))$
  return c

For $n = 1$ or, more generally, if $m_i = m_1$ holds for all plaintexts m_i, then the polynomial degenerates into a straight horizontal line and thus to $c = symEnc_{k_1}(m_1)$, which is equivalent to a normal symmetric encryption scheme.

Randomized Implementation Even though a randomized multi-encryption can be reached by simply adding public but random salt to the hash-function, in this case, adding some additional random tuples (r_i, s_i) from the randomization space $(\mathcal{K}, \mathcal{M})^q$ with $r_i \in_R \mathcal{K}$ and $s_i \in_R \mathcal{M}$, will change the shape (coefficients) of the resulting ciphertext c completely. Additionally to the randomized shape, the true amount of ciphertexts in use can be hidden if the amount q of tuples added is ephemeral as well.

Function $\mathsf{rEnc}_K(M, R)$
Input: $M = (m_1, \ldots, m_n)$, $K = (k_1, \ldots, k_n), R = ((r_1, s_1) \ldots, (r_q, s_q))$
begin
  for $i \leftarrow 1$ **to** n **do**
    $k'_i \leftarrow Hash(k_i) \mod p$
    $m'_i \leftarrow SymEnc_{k_i}(m_i)$
  for $i \leftarrow n+1$ **to** $n+q$ **do**
    $k'_i \leftarrow Hash(r_i)$
    $m'_i \leftarrow symEnc_{r_i}(s_i)$
  $c(x) \leftarrow \mathsf{Lagrange}((k'_1, m'_1), \ldots, (k'_{n+q}, m'_{n+q}))$
  return c

6.4.3. Decryption

The following algorithm describes the basic decryption function $\mathsf{Dec}_{k_i}(c)$. No matter how the multi-encryption c has been constructed, the decryption always works the same way:

Function $\mathsf{Dec}_{k_i}(c)$
Input: c, k_i
begin
  $k'_i \leftarrow Hash(k_i)$
  $m'_i \leftarrow c(k'_i)$
  $m \leftarrow symDec_{k_i}(m')$
  return m;

6.5. Analyzing the Implementation

In the following, the given implementation is analyzed for efficiency and security. Finally the implementation is compared to the requirements given in Section 6.2.

6.5.1. Efficiency

Considering the field operations of $\mathbb{Z}_p$, i.e., addition, subtraction, multiplication, and division modulo p, as primitive operations, $O(n^2)$ many operations are needed to compute $\mathsf{Enc}_K(M)$, i.e., $O(n^2)$ for polynomial interpolation. To compute $\mathsf{Dec}_{k_i}(c)$, $O(n \log n)$ operations are required, i.e. $O(\log n)$ operations for every term in $c'(x)$. Therefore, the running times can be summarized by saying that constructing a ciphertext is quadratic and decrypting a plaintext is quasilinear in n. Note that n will be rather small in most applications. However, it can be chosen as big as required, depending on the field of operation.

6.5.2. Security

Starting the security discussion with a multi-encryption scheme of order $n = 1$, the adversary faces an attack on the used encryption scheme (e.g. AES).

Augmenting n, the adversary can get hold of a set Y with the size $|Y|$ of about $\lfloor (1 - \frac{1}{e})p \rfloor$ elements of the form $symEnc_k(m)$; p denotes the size of the chosen prime field $\mathbb{Z}_|$ and e denotes the Euler constant. This set Y can be obtained by inverting the polynomial represented by c to c^{-1}. The size of this set is due to the following properties:

The polynomial c maps x to y whereas $x, y \in \mathbb{Z}_p$. Observing two specific elements $x_i, y_j \in \mathbb{Z}_p$ the probability q that c will map x_i to y_j is at $q = \frac{1}{p}$, if c is generated at random. Thus, the probability, that x_i does not map to y_j is $q = 1 - \frac{1}{p}$. As each x maps to a y with the same probability, the probability that a certain element y_j is not mapped by any element x results in $q = (1 - \frac{1}{p})^p$, which converges to $\frac{1}{e} = \lim_{p \to \infty} (1 - \frac{1}{p})^p$. This implies that the probability that a specific y_j is mapped by at least one element x is given by $1 - \frac{1}{e}$. Hence the polynomial c behaves in its range like the random function used during its creation.

The fact just described reduces the space of valid encryptions by approximately 37%. Even though this states a certain information leakage, in the case of JCJ05 the attacker is left without a possibility of challenging the remaining 63% encryptions.

Known Plaintext Attack: Concerning known plaintext attacks, the attacker's best strategy is to focus on the underlying cryptographic primitive, (e.g. AES). With an offline brute-force password attack the chances of success depend on the password based key derivation function in use (i.e. key-stretching).[1, 65] However, this attack is not of interest within the JCJ05 context.

True Amount of Keys Used: In the deterministic case, the adversary can extract the amount of keys $|K|$ used during the creation of the multi-encryption ciphertext. This is due to a property of the Lagrange-polynomial the ciphertext

is a representant of: The degree of such a polynomial is directly determined by the amount of sample-points $|K|$ used for its creation. This states a serious contradiction to the claims made in Section 6.2 about quantity-hiding. The claim only holds in the randomized variant. In that case the adversary gains knowledge about $|K|+|R|$, where the size $|R|$ is random. This way, the adversary knows the amount of sample points used during the creation of the ciphertext. This hints about the maximum amount of keys (passwords) possibly managed by a specific multi-encryption ciphertext. But the adversary does not learn the true amount of keys used and hence the creator of such a ciphertext can always safely deny amount claimed.

Deniability: Having a multi-encryption c of order n at hand, the voter is not able to prove to the adversary how the ciphertext was built. This is due to the fact that any n sample points can be used to recreate the very same ciphertext c. An operation that can be done universally and hence does not present a true commitment. Within the context of JCJ05, this can be considered fully deniable.

6.5.3. Properties

In the following, the properties of the given implementation are compared to the properties defined for a multi-encryption scheme:

Multiple Plaintexts: The implementation allows to manage an arbitrary amount of plaintexts independently providing a function mapping the given key to its specified encryption.

Plaintext Hiding: The implementation does not reveal any further information about the plaintexts it manages, than the cryptosystem it maps to. The mapping function itself does not carry any information about the plaintext.

Quantity Hiding: This property is only valid for the randomized implementation. However, the randomized implementation cannot hide the maximum amount of plaintexts it could manage, but it does not revile anything about the exact amount of plaintexts it manages.

Independence of Keys: The implementation does provide a certain amount of information about the possible ciphertexts in use (37 %), however there is no information extractable leading to the keying material.

No Search: This implementation allows to retrieve a plaintext by simply applying the genuine key to the created polynomial and its resulting ciphertext.

Indistinguishability of Keys: Applying a false key to the created polynomial and hence to the resulting ciphertext will result in an arbitrary plaintext (usually of high entropy). Hence this implementation only works for plainttext material of high entropy (such as the credential keys provided by a JCJ05 derivative).

Reusability of Keys: This implementation only provides this property if a public but random salt is applied to the hash-function per creation.

6.5.4. A Thought on Security Within the Context of JCJ05

Coming back to the subject of JCJ05, security is not affected negatively by the use of a multi-encryption scheme and only depends on the cryptographic primitives in use. As the plaintext consists of possible credentials only and hence is of high entropy and as the underlying system (an implementation of JCJ05) is not challengeable, the adversary is left with a brute-force attack without any signaling of success. This implies that within the context of JCJ05 the security of a multi-encryption scheme does not depend on the entropy of the chosen passwords. Please note that even though this conclusion is correct as it stands, it simply means that the adversary can neither learn any secret stored, nor any key used within a particular multi-encryption ciphertext.

6.6. Conclusion

Within this chapter, the need and the required properties for a multi-encryption scheme have been introduced. The provided implementation states a true instance of such a multi-encryption scheme if it is applied in a special setting, where the plaintext-material consists of messages that contain high entropy. This is exactly the setting provided by JCJ05 schemes.

Chapter 7

Rendering JCJ05-Schemes Manageable for the Voter: A Hopeless Quest

According to the previous chapter, an implementation of a coercion resistant democratic *e-voting system* has to be described as the composition of two subsystems, namely the *JCJ05 subsystem* implementing the JCJ05-scheme and the *password subsystem* implementing a cryptographic component (and a simple physical helper device). This chapter presents a new aspect of verifiability and coercion resistance and provides evidence that proves the original JCJ05 to be broken by design in the context of *human* voters. More precisely, it is demonstrated that the JCJ05-scheme (including the linear board flooding prone derivatives), cannot be implemented in an e-voting system usable by human voters without violating the claimed coercion resistance or lowering the claimed individual verifiability. Within these schemes, the human members of the electorate will never know if their true intention really made it to the final tally (counted as intended).

7.1. The Voter

In the following, neither the creation of the ballot nor the act of bringing it into the voting system are of special interest and therefore will be omitted to simplify the argumentation.

7.1.1. Experiences on the Voter's side

The main focus within this part is set on the following voter experiences:

Entering the key: The voter enters some kind of password (could be anything manageable by the human brain). Internally this reveals the high entropy credential used by the underlying system (usually involving some sort of key derivation). Depending on the entered password, either the valid credential σ is unveiled, or some fake credential is unveiled.

Verification: The voter checks the response of the system for a specific password entered.

7.1.2. The Human Voter Model

Throughout the literature concerning the JCJ05-scheme, the voter has not really been taken into account. The model in use in JCJ05 assumes that the human voter is able to perform every task which is computable, and to process arbitrary data in a perfect manner, and has therefore been modeled very strictly, as *"yet another perfect universal Turing machine"*. Within this dissertation, the human voter is shifted a bit more into focus and thus the model representing the human voter has to be adapted with a "tiny" *relaxation*: Human voters tend to be rather ineffective when it comes to the processing of high entropy data [15, 43, 46]. This indicates that the human voter cannot simply be modeled as a universal Turing machine.

Definition 7.1.1 (Relaxed Voter Model) *The relaxed voter model reflects the possibility of unintended and unrecognized alteration during processing of high entropy data. The model is realized as a universal Turing machine with noisy data channels, whereas the level of noise introduced is directly linked to the entropy level of the data in use: The higher the entropy of the data to be processed, the noisier the involved channels.*

Even though the relaxed voter model lacks an explicit entropy to noise ratio, and is not based on explicit scientific deduction, it serves as a more realistic model throughout the rest of this dissertation. In the study "Of Passwords and People" [71] one finding that heavily supports the relaxed human voter model, was that even participants remembering their passwords needed on average 1.22 attempts to log into a challengeable test system, hence the noise to be reflected by the model is not on the small side.

7.2. The System's Response

In JCJ05, the human voter chooses exactly amongst two choices when entering the password: Either the human voter wants to express the right to vote, or the voter wants to fake it. The e-voting system will always respond in the same way —it will

accept. But how can the human voter be convinced by the e-voting system that the entered password really reflects the human voter's intention? What if the human voter just made a typo (=noisy channel)?

In 2008 Clark and Haengartner presented their study of *panic passwords* [27] as a credential management scheme to remote e-voting. Their threat-model consists of four pragmatic assumptions:

Kerckhoffs's Principles: Everything but the shared secret (password(s)) between the voter and the system is considered common knowledge and hence known to the adversary.

Observational Principle: The adversary faces a semi-private channel from the voter to the e-voting system and thus is able to observe any action on the voter's side during the coercive act. The adversary is even able to actively replace the voter after having coerced the voter into handing over the password.

Iteration Principle: The adversary is not limited to a single instance of coercion against a specific voter. This way, the adversary can force the voter to authenticate multiple times. In combination with the observational principle, the adversary may force the voter to use a different password each time.

Forced-Randomization Principle: The adversary can freely choose any strategy through the order in which the voter has to apply the passwords. For example, the adversary can force the voter to commit to a set of passwords prior to its use, so that the adversary can randomly choose the order in which to apply them.

Furthermore Clark and Haengartner described 4 threat parameters that in general can be applied individually. However, two of them are clearly defined within the context of coercion-resistant remote e-voting and hence they are fixed within this dissertation:

Adversarial Persistence: The period of time a concrete attack takes place.

System's Reaction and Response: After having received a panic password, the system has three options. It can perform an unobserved reaction but not change the observable response, it can alter the response without committing any unobserved reaction, or it can do both.

Adversarial Goal: The fix goal of the adversary is to either hinder the voter from voting or to coerce the voter to vote for a specific choice.

Screening vs. Signaling: In the fixed context of coercion-resistant remote e-voting it is not possible for the adversary to screen (distinguish) the entry of a panic password, nor is it possible for the voter to signal (prove) the entry of the correct password.

7.3. Panic Passwords and JCJ05

As mentioned before, the e-voting system always accepts any password (credential), but only in the case where the correct password is entered will the system count the vote, otherwise it rejects the vote silently, a behavior defined as unobserved reaction. It seems as if each human voter only has to remember one password. Any other password acts as a panic password. But as described by Clark and Haengartner, there is a non-negligible chance of an unintended use of a panic password by simply misspelling the proper password. As neither the human voter nor the adversary is able to screen the correct password (distinguish the response), the human voter cannot individually verify if the vote has been counted and hence correctness – stating a core requirement of democratic voting– is violated. In order to solve this problem, the authors state that either the system has to be typo tolerant in some way, or the panic passwords have to be limited in a way that only passwords with a certain Damerau-Levenshtein distance [8] (the minimum number of a specific set of operations to transform a string into another one), are acceptable passwords. This is where their paper ends and where our contribution starts.

7.3.1. Naïve Approach

Starting with the simplest idea of using a simple symmetric cryptographic storage, such as an AES-ciphertext for the password subsystem, the only thing the voter has to memorize is a freely manageable key (password). An important aspect of this approach is that no matter which key is entered to decrypt the ciphertext, the resulting message will always be of high entropy and accepted by the underlying voting system. This conforms well with the requirement that the voter should not be able to signal the correct password. In this context, the system is quasi deniable, as the underlying e-voting system's response is always unobserved, and hence not challengeable at all. However, the entry of the key by the human voter is error prone —due to the noisy channel between the voter and the credential management system. As the underlying implementation of the JCJ05-scheme does not provide any typo resistance this task has to be done by the password system in use. Approaching the problem by introducing any kind of error correction such as spell-checking, would dramatically reduce the key space. This way the e-voting system would allow a dictionary attack.

7.3.2. Multiple Ciphertexts

The conclusion of the naïve approach implies that the password subsystem must signal some kind of voter-dedicated error detection, where the error detection cannot be made public, hence neither allows the voter to signal nor the attacker to screen the system's response.

2-ciphertexts: As a first solution, the password subsystem requires two symmetric-encryptions of the very same JCJ05 credential σ using different keys. This way, the credential management system only releases the credential if the decrypted messages are identical. This rules out the problem of typos but of course completely breaks the coercion resistance, as symmetric-encryption systems with small keys (such as AES) show a strong collision resistance they do not allow to find arbitrary plaintext-collisions. This way an attacker can force the voter to hand over two different keys that will reveal the very same plaintext.

2(N + 1)-ciphertexts: To solve the problem just introduced, the natural extrapolation is to create $N + 1$-pair of ciphertexts, one pair encrypting the correct credential whereas the other N tuples encrypt different credentials, hence serve as panic-passwords. Unfortunately this approach gives hints to the adversary about the amount of passwords the human voter manages and therefore the adversary can simply urge the voter to spend them all. This way the problem remains. Hence there must be a way that allows the human voter to hide the amount of passwords in use.

2(N + 1)-ciphertexts using a multi-encryption scheme: With a multi-encryption scheme and its concrete implementation at hand, the voter is able to manage multiple password-tuples for multiple credentials. This gives the system (including the device handling the credential) two responses:

Accept: The system accepts the password(-pair)

Reject: The system rejects as the password(-pair) does not match

As described in Chapter 6, using the deterministic implementation, the original problem persists: The adversary can extract the amount of passwords in use and can therefore force the voter to spend them all.

This implies that the human voter has to use the randomized variant of the multi-encryption implementation. This way the voter creates a list of messages M in order to create a multi-encryption for the following strategies:

Deterministic Match: $(\sigma, \sigma, a_1, a_1, \ldots, a_n, a_n)$ whereas a_i is some bogus[1] value within the same space as σ. Furthermore the voter creates a K consisting of distinct elements $(pa_0, pb_0, pa_1, pb_1, \ldots, pa_n, pb_n)$ whereas pa_i, pb_i are considered to be some passwords bearing a certain minimal-entropy. Finally, a certain set R is added to the encryption-function resulting in a perfectly deniable c within the context of JCJ05.

If the password(-pair)s used by the human voter shows enough distance to the passwords used for faking, the human voter can be sure that the passwords reflected the intention into the voting system.

Probabilistic Match: $(\sigma, \sigma', a_1, a'_1, \ldots, a_n, a'_n)$ whereas a_i is some bogus value within the same space as σ. Ensuring σ and σ' are different but congruent modulo n where n is some public small value (The same is true per tuple a_i, a'_i), allows the challenging to be probabilistic in nature. Furthermore, the voter creates a K consisting of distinct elements $(pa_0, pb_0, pa_1, pb_1, \ldots, pa_n, pb_n)$ whereas pa_i, pb_i are considered to be some passwords bearing a certain minimal-entropy. Finally, a certain set R is added to the encryption-function resulting in a perfectly deniable c within the context of JCJ05.

Entering two passwords pa_i, pb_i will result in two values congruent modulo n. If the password(-pair)s used by the human voter shows enough distance to the passwords used for faking, the voter can be convinced with probability $\frac{n-1}{n}$ that the passwords reflected the intention into the voting system. However, with a probability of $\frac{1}{n}$ this mode brings forth the answer of a *false reject*, where a vote seemingly attached with the correct password will be rejected silently.

7.3.3. Security

Applying the $2(N+1)$-ciphertext strategy renders the multi-encryption and therefore the whole JCJ05 system challengeable. The adversary is now able to search offline for *collisions*, passwords resulting in the very same ciphertext and plaintext respectively.

Deterministic: If two somewhat low entropy passwords match, an entry of a $2(N+1)$-ciphertext has been found with very high probability. As N is usually a rather small number, the probability of having found the one ciphertext invoking the right to vote is thus very high. This raises a clear security issue and requires the passwords to carry a minimum amount of entropy equivalent to the stored high entropy credential in order to withstand a multi-

[1] Reminder: Bogus $\mapsto$ Unknown to the e-voting system.

year distributed brute-force password attack; which brings us back on "field one".

Probabilistic: If two somewhat low entropy passwords match in the sense that they produce values congruent modulo n, a $2(N+1)$-ciphertext has been found with a probability of $\frac{n}{|\mathcal{K}|}$ where $|\mathcal{K}|$ denotes the order of the key space, which is of somewhat low entropy. Please note, that even though in theory $|\mathcal{K}|$ is unbounded, in reality it is not, as it is used by humans; in fact it is usually extremely small. According to a large-scale study from Florêncio et al. [44] the $|\mathcal{K}|$ is somewhere between 2^{40} and 2^{60} for the average human[2]. Augmenting n reduces false rejects but at the same ratio lowers the entropy of the key space, and thus serves the voter and the adversary—stating a true dilemma. Moreover, our internal studies (including myself) showed that the passwords used as a tuple most often show a strong relationship and hence are not independent enough to withstand a password-attack.

For the best of my knowledge there is no deterministic cryptographic possibility to render the original JCJ05-scheme usable for the human voter. In the probabilistic case however, the key space which is already on the small side gets even smaller, and most certainly can be broken by the offline attacker using social intelligence.

[2]Currently (2012) a secret secured with a private key of less than 90 bits of entropy ($< 2^{90}$) is considered to be highly vulnerable to a brute-force attack [76].

Chapter 8

Rendering Board Flooding Resistant JCJ05-Schemes Manageable for the Voter

The previous chapter ended by proving that JCJ05 cannot be used to provide the targeted coercion-resistant democratic *e-voting system* when the unrealistic perfect voter model is replaced by the relaxed voter model.

Within this chapter, the relaxed voter model is applied on board flooding resistant JCJ05-schemes presented in Part I. If combined, can any of those schemes remain an *E2E-verifiable* voting scheme without loosing the claimed coercion-resistance?

As the investigations on board flooding resistant JCJ05-schemes only started during this dissertation, the special needs for the voters had yet to be explored. In particular, the resistance to board flooding introduced in Part I requires the system to manage more than one password at the voter's side, which again shows some negative influences on noise level in the relaxed voter model.

8.1. General Security Aspects of the System

The board flooding resistant JCJ05-subsystem will not accept arbitrary credentials any more. Instead of the two reactions in the JCJ05 case, the e-voting system now publicly rejects invalid passwords. In order to allow fully deniability with respect to the whole system, the human voter now has to be able to invoke every possible response of the system at will.

In particular, the human voter needs to be able to invoke the following responses:

Accept: The system accepts the password(-pair)

Reject I: The system rejects the password(-pair) because it has already been used

Reject II: The system rejects the password(-pair) because it is a bogus password

8.2. Security Aspects in the Generic Approach

The generic approach requires the human voter to apply passwords for different semantics. One is required to express the will to execute the right to vote, just as it is the case in the e-voting system using JCJ05, and (multiple) others are required in order to be allowed to post votes.

8.2.1. On the Voter's Side

The combination of the multi-semantics of the passwords, and the requirement that the human voter has to be able to apply passwords so that all of the responses mentioned can be invoked – without spoiling the right to vote – requires quite some human voter training and will therefore most likely not be accepted by the human voter, thus cannot be considered as usable.

8.2.2. Partial Typo Resistance

In complete contrast to the e-voting system using JCJ05, the e-voting system using the generic approach provides an intrinsic typo resistance. Unfortunately, it is only applicable to the posting tickets. This is due to the different responses the board flooding resistant system provides. If the human voter correctly enters the password for a posting ticket the system will respond with an *Accept* or with a *Reject I* if the password has already been used. On the other hand, if the human voter misspells the password for a posting ticket, the system will respond with a *Reject II*. Therefore, there is no need for any further typo-resistance strategy on the voter's side and the offline attack is defeated.

Unfortunately though, this intrinsic typo resistance is not given for the credential σ used to mark the will to vote. There, the system behaves the same way as the original JCJ05 and so does the voter's experience: by applying the $2(N+1)$-ciphertexts strategy to the password subsystem the attacker can find the credential offline via a brute-force attack, and urge the voter to hand over a single valid posting ticket.

Therefore this partial typo resistance is no better than the e-voting system using a JCJ05-subsystem and has to be considered as broken by design if applied on the relaxed voter model.

8.3. Security Aspects in the Integrated Approach

The integrated approach already shows a somewhat weaker form of coercion resistance. Surprisingly though, it is the only approach where the introduction of the relaxed voter model does not break the system.

8.3.1. On the Voter's Side

In contrast to the generic approach, the integrated approach provides a much better usability for the human voter, as the required passwords all bear the same semantics: the will to execute the right to vote. Again, the human voter has to be able to apply passwords in such a way that all mentioned responses can be invoked, without affecting the right to vote. But, this time the integrated approach lends a helping hand.

8.3.2. Full Typo Resistance

In the integrated approach, the system provides fully intrinsic typo-resistance, as the system's responses only depend on the entered credential. If the voter correctly enters the password for a credential of any flavor (σ for the true intention or τ for a fake intention), the system will respond with an *Accept* or with a *Reject I* if the password has already been used. If the voter misspells the desired password, the system will always respond with a *Reject II*. This states *the* true advantage over all the other schemes and allows to drop any typo-resistance strategy whatsoever on the voter's side.

So, the integrated approach states the *only* E2E-verifiable coercion-resistant remote e-voting scheme when the relaxed voter model is in use. The original JCJ05-scheme requires the perfect – non-human – voter in order to show the E2E-verifiability property, a constraint which is unrealistic.[1]

8.3.3. Concerning the Password Strength

The system using the integrated approach still remains attackable. But it is restricted to a (distributed) online attack. If the system is built in a way that only a

[1] Modeling the human without the 'right' to make an unintended misstake seems rather taunting (to me as a human).

fixed maximum amount of u password responses is accepted during a voting period, the minimum entropy requirements for the passwords only depends on u and a security parameter r denoting the probability $p = \frac{1}{2^r}$ that a password can be guessed and thus results in $\log_2(u) + r$ bits. A numerical example with slightly superimposed numbers shall demonstrate the requirements for the brain of the human voter:

Assuming a security parameter of $r = 20$ and a maximum amount of $u = 2^{40}$ postings for a voting period, the minimum required entropy for a password results in $\log_2(2^{40}) + 20$ and thus is at about 60-bits. By using an alphabet of 64 characters (2^6), about 10 independent characters are needed in order to gain enough entropy to withstand the online attack.

8.4. Concerning Passwords

Many papers have been written, concerning passwords [44, 71, 111] and there are literally billions of web-sites providing 'good' advise on how to create- and remember them. However, humans keep struggling to manage well chosen passwords. A group of our students [77] implemented a first variant of a multi-encryption scheme. To ease the use of the implementation, their program provided sensor data available by today's smart-phones. This way the input possibilities for the generation of passwords could be enriched. Even though on a theoretical level, this does not augment entropy by taking the physical context into account human users are able to manage passwords with somewhat higher entropy in practice. This was one of the findings of their work during the usability studies. In their work, they allowed different physical aspects into the system, so, that the human user only had to decide if the actual sensor data should become part of the password. For example, the voter creates a password by taking the magnetic compass into account and so provoking a panic password if facing north, while facing south would have released the true credential. In the desperate search for manageable entropy by humans every bit counts.

Physical Device: It is important to emphasize that this particular implementation of theirs should not and cannot be considered as a possible solution. The smart-phones in use cannot be proven trustworthy by definition, as they propose the equivalent of a universal Turing machine. As mentioned before, this implies that the question of non-existence of malicious software can never be answered positively. Therefore their work demonstrated, that the serious management of a multi-encryption scheme can be realized in a way that is usable for humans, but that a trustworthy physical device is urgently needed.

8.5. Summary

JCJ05 as described in the literature cannot be brought to reality as the underlying voter model is too strict. As almost all of it's derivatives it requires the perfect voter when it comes to credential processing. The only scheme that is compatible with the relaxed voter model is the *Integrated Approach* described in Section 5.1.2. This scheme implicitly provides full error-detection on the credential handling on the voter's side. In combination with the introduced multi-encryption scheme it has the potential for a usable realization. As a side effect, it also provides hints on credential-attacks, when an adversary starts to challenge the system in order to break a multi-encryption ciphertext, as this attack produces a significant amount of bogus credentials that will be rejected by the scheme. These findings conclude, that among the listed possibilities, this combination is the only one worth considering when planning to implement a usable system.

Part III.

Secure and Private Voting on Adversarial Ground

Chapter 9

Towards a Usable Solution of the Secure-Platform-Problem

The protocols described in Part I and the cryptographic component introduced in Part II require computational aid for the human voter. The introduction of the secure platform problem in Section 2.5 described the inherent risk of hosting the adversary if the devices in use propose the equivalent of a universal Turing machine. To emphasize this again, this implies that computer equipment such as PCs or smart-phones cannot be trusted by definition no matter how thoroughly they are analyzed. All these findings naturally ask for a dedicated trusted device, not allowed to accept any new programming instructions, only executing pre-loaded and analyzable program code. But in order to be usable, the computational aid has to address the human voter and thus has to present the operations with a certain appeal, following the Zeitgeist—which seems contradictory after all.

Within our publication [53], originally approved and published by the Swiss Federal Chancellery, and our peer reviewed chapter in [54], a solution is presented for the secure platform problem (including the cryptographic requirements on the voter's side), applicable to the described dilemma. Additionally, the described solution has been implemented as a Bachelor thesis by Pellegrini & vonBergen [88] in order to study the practical aspects of the stated solution.

9.1. State of the Art

Throughout the remote e-voting literature, there exist several approaches involving dedicated trusted devices. In order to relax the unacceptable requirements of the untappable channel in HS00 [58], Magkos et al. [78] as well as Lee et al. [74, 75] propose a secure hardware device called a tamper-resistant randomizer (TRR)

replacing both, the trusted third party randomizer and the untappable channel. To handle the credential and the cryptographic requirements on the voter's side of JCJ05, Schweisgut [101, 103] proposes a hardware device called an observer. Meister et al. [80] propose the use of the European Citizen Card (ECC) for the same purpose. So, the idea of relying on special hardware on the voter's side is not entirely new. In fact it does not stop at the theoretical level, but is in use in Estonia, where smart-cards take care of voter authentication [16]. But, none of those approaches is meant to solve the secure platform problem.

On the other hand, Joaquim et al. describe a protocol [117, 118], directly addressing the secure platform problem by proposing a combination of a special hardware device and another type of remote e-voting called code voting [87]. A special card reader, a FINREAD reader [1] with a small LCD display (4 lines of 20 characters each), is proposed to display the ballot, where the provided keypad serves as the channel to read the voter's candidate choice. A personal smart-card provides the public-key authentication. However, this approach cannot be considered as 'user friendly' due to the restricted usability of the small display and the simple keypad.

To provide privacy even in the presence of strong malware, we propose that the voting device receives its information from the voter's insecure platform in form of matrix barcodes. In 2002 Clarke et al. [17] already presented a camera-based authentication protocol, in order to establish a so called "Unidirectional Authentication with Secure Approval Channel" (*UASAC*) [17], where the human user can approve the message presented by a untrusted computer with rich interface using a camera equipped personal trusted device that authenticates the presented information.

9.2. Combined Approach

The solution proposed in this chapter combines a device similar to the FINREAD reader in terms of I/O capabilities, and similar to the device described by Clarke, thus allows the voter to use the rich interface of the personal platform to do the voting. This combination finally unites usability and a secure platform. However, this solution comes at a monetary price.

In our original publication [53], the solution (and the resulting implementation) was embedded within a remote e-voting protocol not comprising coercion resistance. Within this dissertation however, the proposed solution is combined with the protocols and components described in Parts I & II. Interestingly though, this reduces the amount of required trusted hardware to a single device with no inner

[1] FINREAD: European card reader standard for secure financial transactions and electronic signatures on open networks

secret to protect (no private key). Nevertheless, the core idea for the solution of the secure platform problem stays the same: To combine a trusted device with limited computational capabilities with the voter's untrustworthy but powerful computer equipment with strong abilities when interacting with humans.

9.2.1. Voting Platform

The main idea of E2E-verifiable remote e-voting is to provide human voters the ability to cast and verify the vote in a known and accessible environment—at home. Therefore, the equipment required for remote e-voting has to be available and usable individually by all members of the electorate. Bringing in economical aspects, it becomes obvious that the human voter is best served if it is possible to achieve the voting act by using personal equipment (="Bring your own device") as a *voting platform*. As the personal equipment used is very heterogeneous throughout the electorate, the representation on the different devices has to be unified in such a way that there is no special treatment required in order to run the voting software. But still, the whole voting process on the voter's side has to provide a certain experience so that every voter feels safe and understood. Certainly, no cryptography whatsoever is allowed to shine through and all must come down to the action recognizable and expected by the voter—to select. Surprisingly though, the only thing not doable on the voting platform is the possibility to mark the final volition. If the voter would do so on the voting platform, the adversary 'within' would immediately know.

Channels

The task of the voting platform is 'to serve and to please' the voter. As a terminal of remote e-voting, the voting platform has to be connected to the Internet in order to communicate with the rest of the e-voting system. On the other hand, all suitable channels to the human voter have to be used in order to communicate in a convenient way. In [53], the computer screen plays a dominant role. It establishes a broadcast channel to the human voter as well as to the trusted device and broadcasts a superposition of all possible voting options (see Figures 9.2, 9.3). As long as the voting platform cannot learn which option has been chosen by the voter, this superposition remains intact. This exact property can be achieved by using a trusted voting device.

9.2.2. Trusted Voting Device

The setting this dissertation is operating in requires a single trustworthy hardware token on the human voter's side—the *trusted voting device*. Again, it is an absolute must, that this device only executes a finite (hard wired) set of algorithms and does

not execute any further instructions. It is this reduction to a push-down automata, that theoretically allows to completely analyze the device, a property providing the basis for trust.

The device has to be certified and tamper proof in such a way that the human voter can *recognize* if the device has been physically altered. This way the human voter can build trust (regarding privacy) in the *layer of experts* having analyzed the device. The functioning of the device can be verified individually by establishing test environments not distinguishable from the true voting environment by the device. This allows the human voter to establish trust regarding correct functioning.

Even though the trusted voting device has to be tamper proof, it does not hold any secret, hence is not personalized. This loose coupling thwarts vote abstention attacks, as the adversary cannot hinder the human voter on a large scale from using any trusted voting device, without being noticed.

Channels

The trusted voting device serves as a trustworthy translator between the human voter and the rest of the voting system. Its restricted nature does not allow high channel capacities. In [53] the human-compatible channel consists of a simple 2-lined text display allowing the device to send messages to the human voter. Communication in the opposite direction is achieved through a simple keypad, so the human voter can type in context dependent decisions. The channel to the rest of the system enables the trusted voting device to communicate with the remaining parts of the voting system. In order to receive information from the voting system, the trusted voting device possesses a matrix-code scanner. This way, the voting platform can broadcast matrix-codes on the screen, readable by the trusted voting device. To send information to the voting system, the trusted voting device has the ability to write out *encrypted* data as a file via a computer connection (such as Universal Serial Bus).

Unidirectional Broadcast Channel

The key to establishing privacy on the voter's side is given by the implicit unidirectional broadcast channel controllable and verifiable by the human voter. In order to make a selection, the human voter targets the matrix-code scanner of the trusted voting device towards the corresponding matrix-code presented at the voting platform (where the measuring destroys the superposition of all possible options for the trusted voting device). This combination of targeting and reading allows the human voter to express the selection very naturally to the trusted voting device, which then presents the content of the received data to the voter. The voter then verifies the coherency of the selected option and if approved tells the trusted voting

device to act accordingly. Please note that the voting platform will never learn about that act.

9.2.3. Modes of Operation

Within the context of a coercion-resistant remote E2E-verifiable e-voting system, the tasks for a human voter are manifold but often comprise cryptography. This is where the trusted voting device has to serve the human.

Credential Management

The trusted voting device has to support the human with an implementation of a multi-encryption scheme.

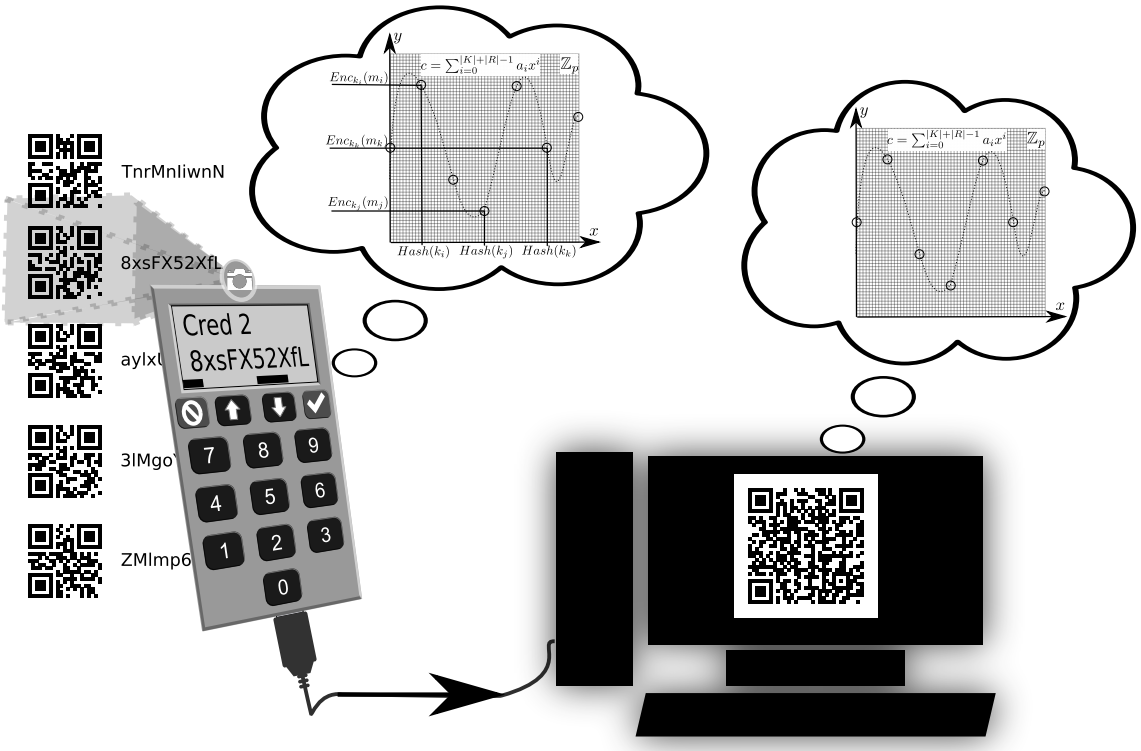


Figure 9.1.: Device reading in all credentials and passwords in order to create a multi-encryption ciphertext. This ciphertext can then be announced publicly so neither the voter nor the device has to remember the ciphertext.

During the setup phase, within the untappable channel, the human voter can create the multi-encryption ciphertext by reading in the credentials and

corresponding passwords with the device. The device then creates the freely publishable multi-encryption ciphertext as a file, that can be read out safely via any computer connection. This ciphertext can even be put on the public bulletin board, so neither the voter nor the device has to manage it. The only thing the voter takes out of the untappable channel is the passwords... 'buried' in the voter's brain. (See Figure 9.1)

During the voting phase, the human voter can read the voter's specific ciphertext back into the device. This enables the voter to manage the multi-encryption ciphertext via the trusted device.

Ballot Management

The trusted voting device has to support the human in verifying digitally signed voting material coming from the authorities. This way the voter can be sure to make the selection based on the correct voting material that has not been tampered somewhere between the server and the display of the untrustworthy personal computer. To preserve full voter privacy, it must be possible to target a selection with the trusted device. Voter privacy results from the trusted device being capable of reading the targeted selection without giving any feedback to the voting platform presenting all the options on the screen. (See Figures 9.2, 9.3)

9.2.4. Randomness Within the Trusted Device

As already mentioned in 9.2.2, the trusted voting device must not protect any secret in order to be used within the context of coercion resistance. In contrast, it must be possible to completely analyze the device. This includes the randomization required for the probabilistic encryption. As mentioned in Section 2.4, randomization is ephemeral, meaning extremely volatile in nature. Hence, the trusted voting device must reflect that fact by constantly changing the internal randomization. But as the device is not made to protect any secret, it is not allowed to base randomization solely on a pseudo random generator, as there is always an initialization vector involved which might be extracted. Instead, the device must have access to a true random generator which is fed by entropy, hence not completely manipulable by the owner. This can be achieved by gaining Johnson-Nyquist noise [86] using the noisy sensors of the device (e.g. CCD-sensor of the camera) as a true source [39] of entropy. One should not underestimate the quality of such a source, as the noise is provoked on the quantum mechanical level and therefore even works if the lens of

the camera is completely blocked. [2] Even if the source is biased, it is possible to gain proper randomness from it by applying well studied techniques such as described by Barak. [7] Analyzing the device (expert layer) it is verifiable if this random source is in use. In order to give the human voter reason to trust the functioning of the constantly changing randomization, parts of it can be made perceptual at the audio level (loudspeaker) for example. If the audio representation does not happen to be equivalent to white noise, but provides some clear pattern, the random source can be considered as faulty. Please note, that this method is not accepted as an instrument for measuring the quality of a random source, but at least it gives the human voter a hint as to whether the random source is stalled.[3]

This way, the randomization cannot be forced from an untampered device, as the device constantly keeps forgetting it.

9.3. Modes of Application

As voting includes elections as well as initiatives and referenda, all modes of application have to be covered by the solution in order for it to prove itself usable.

9.3.1. Initiatives and Referenda

This mode of application can be demonstrated best in scenarios requiring the voter to make a 1 out of n selection, where n is very small (e.g., $n \leq 4$). This way the voting platform provides all possible volitions *at the same time* on the screen.[4] The voter then points at the desired option which will be read by the trusted device. The trusted device then presents the selection again to the human voter, who then can finish the voting process on the device.

9.3.2. Elections

When it comes to large-scale elections with a large m out of an even larger n possibilities to choose from, the mode of application demonstrated above works in theory but fails in practice due to usability constraints. The approach chosen

[2] Experiments demonstrating this fact can be done with an open source software called video_entropyd, written and maintained by Folkert Vanheusden `http://www.vanheusden.com/ved/`

[3] It must be emphasized, that an attack on randomness can only be recognized by the layer of experts during a complete analyzation of the device! It is not possible to find an attack by listening to the random source, the only thing that can be heard is a malfunctioning of the random-gathering process if e.g., the used sensor is not working.

[4] For this scenario it would even be possible to print out all possible volitions without disclosing any information to the adversary controlling the computer.

for this kind of voting requires the voting platform to continuously broadcast the actual selection. If the voter changes the selection, the corresponding matrix code has to change accordingly in real time. This way, the voting platform cannot deduce the true selection made by the voter. However, the quality of this property heavily relies on the voter's interaction and behavior. The following example shows, that it is indeed possible to 'blind' the adversary present in the untrusted platform: Let there be a voter ready to go through the election procedure multiple times. One time, the voter selects the well desired election choices and will submit the vote using the true σ credential. In all other cases, the voter will choose different but realistic election choices, and will submit them using τ credentials to indicate a fake vote or even by using bogus credentials. This way, the adversary within the untrusted platform cannot succeed using a timing attack, where it is assumed that the display-time for a selected election to be cast is different from other selections. One might think that this still gives the adversary quite some information, but if the voters choices consist of contradictory but equal probable ones, the entropy (level of surprise) of the gained data is very low.

9.4. Verification Management

The trusted voting device has to support the human in order to allow individual verification. Hence the trusted device must be able to verify the proofs made by the remote e-voting system concerning the cast ballot.

Figure 9.2.: The voting platform showing an initiative where all possible options are in superposition. The trusted voting device is then used by the voter to select a specific option. The image of the voting platform represents the look of the implementation in [88]

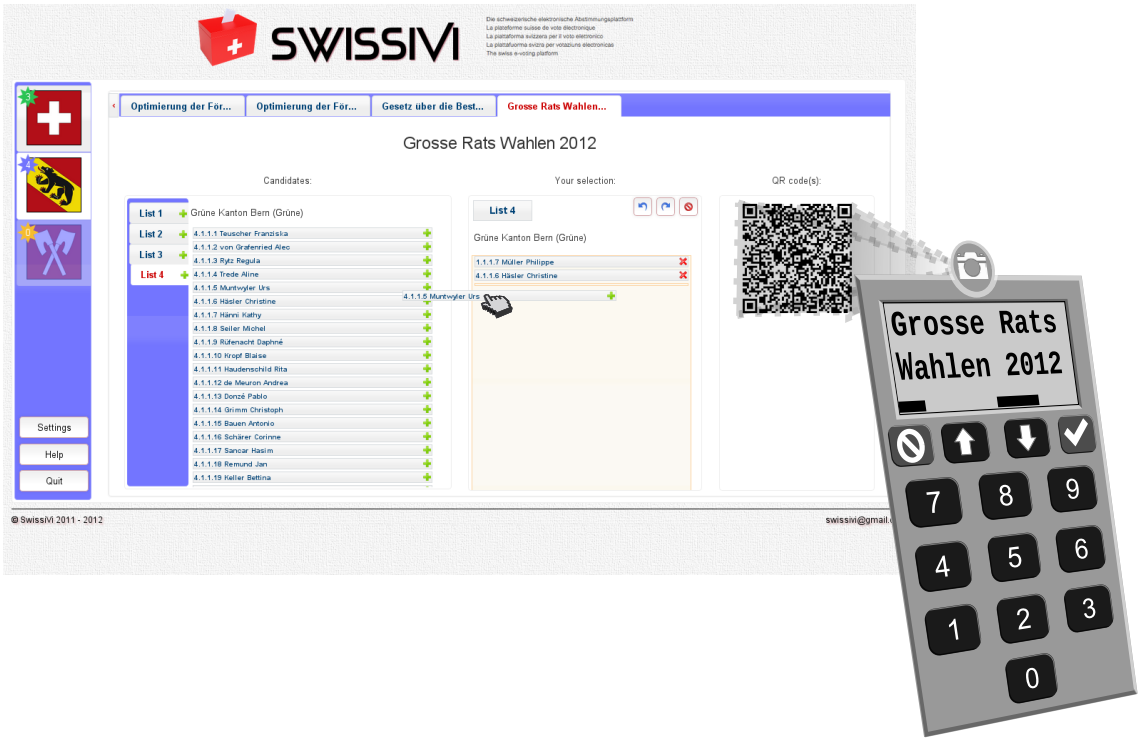


Figure 9.3.: The voting platform presents a highly interactive form of a complex election scenario. The matrix code keeps changing when a new selection is made. This superpositions the possibilities over time. The trusted voting device is used to capture the desired options. The image of the voting platform represents the look of the implementation in [88]

Chapter 10

Putting it All Together

This chapter brings together all the findings and contributions made throughout this thesis, in order to present a final image of how a complete usable coercion resistant E2E-verifiable remote e-voting system can be achieved under the assumption of the relaxed voter model. It has a summary character and does not go into deeper details anymore, but rather gives a last 'fly-over' impression of the scenery built up throughout this work. Please note, that the following description has not been elaborated in practice, and therefore remains theoretical.

10.1. Voting Setup

In order to set up the voting system (Section 5.1.2), the human authorities set up a distributed k-out-of-n thresholded key-pair (pk_{AX}, sk_{AX}) for an asymmetric encryption scheme with homomorphic properties. Each human authority has to safely store away their part of the private key sk_{AX_i}.

The computing devices, on which the human authorities process the private key-parts, have to be trustworthy. As coercion is not considered a problem at this stage, the private keying material can be stored and processed on a physical computing device dedicated to this task only in order to minimize possible side-channels. Please note that the processing of the private-keying material is the Achilles' heel of every voting system; its correct handling cannot be proven, but is solely based on trust.

10.1.1. Establishing a Public Key Infrastructure

In order to separate duties, (in contrast to the description in Section 5.1.2), two distinct types of authorities exist which handle the privacy of the voter. This allows the distribution of the task of maintaining privacy of the vote, thus eliminating a single point of failure in access-control. Furthermore, it allows the use of different

cryptographic systems based on their suitability for the tasks for each type of authority.

Credential Authorities: This set of authorities CA provide the signature key sk_C as a certificate with which information concerning credentials is signed.

Authentication Authorities: The set AA of *authentication authorities* provides the public key pk_A with which the voters will encrypt their credentials.

Tallier Authorities: This set TA of *tallier authorities* provides the public key pk_T with which the voters will encrypt their votes.

Election Authorities: This set of authorities EA provides the signature key sk_E as a certificate with which information and selections of an election or voting event will be signed.

10.1.2. Trusted Voting Device

The dedicated trusted hardware used on the voter's side can be mass-manufactured without the need for special security constraints on the manufacturer's side. During the physical assembly of the trusted voting devices, the tuple $ATE = (pk_C, pk_A, pk_T, sk_E)$ and any other public setup parameters are introduced in the device, so that it cannot be altered without tampering with the device. However, the parameters remain freely readable—do not state a secret.

Building Trust: To build trust in the manufactured device line, each device needs to remain completely analyzable at any time. In particular it must be possible to decompose the complete device in order to verify the physical and logical bases. This act almost certainly leads to the complete destruction of one device. Even though this technique of verification is mainly only for a *layer of experts*[1], it gives the electorate reason to trust in the remaining devices of this particular line. However, the human voter must be able to recognize a tampered device; if not, it opens doors for an adversarial injection of 'tainted' devices.

[1]Term borrowed from David Bismark used at the Swiss E-Voting Workshop 2012

10.1.3. Credential Establishment

As the integral approach (Section 5.1.2) is in use, there exist two flavors of credentials, namely σ and τ, which need to be established and distributed.[2] Therefore, the CA creates a set S of distinct true voting credentials σ encrypted under pk_A and publishes it on the public bulletin board. Additionally, each σ of the unencrypted enumerated set is printed and sealed using secure printing. Secure printing is a technique also used in supervised e-voting systems such as prêt à voter [95] or Scantegrity II [24] allowing to print data in a distributed and hidden way, so that no printer can gain enough information to recreate the secret it is about to print.[3] The credential authority then does the same for T, the set of encrypted distinct fake voting credentials τ. However, the printing differs this time: the set of unencrypted τ's is divided into subsets P_i where $|P_i| \approx 20$.[4] The τ to be printed is drawn randomly from within that subset. The repetition of this process results in a list of τs of size $|P_i|$ where some τs are listed multiple times and some τs of the subset are completely missing with a certain probability. The aim of this process is to obfuscate the true amount of distinct τ's printed. This collection of τs is then sealed as one 'package'. This way, remote e-voting requires a very complex infrastructure. Fortunately though, this is only required once at an early stage of the setup of the e-voting system and is no longer needed afterwards, so its cost can be apportioned to several voting events.

10.2. Voter Setup

Every potential member of the electorate has to authenticate and register. However, the process of authentication and registration must be kept completely secret, so in theory, the voter does this via an untappable channel. In practice, the voter will authenticate in person at a registration office. At latest after successful authentication, the voter gets a trusted voting device, with which the remaining procedure will be executed.

Credential Registering: The voter chooses a single sealed σ-credential. Before destruction of the credential information, the voter memorizes the credential with the help of the trusted voting device. The same is done for a randomly chosen sealed

[2] Even though I elaborated on a verifiable and more secure version for the credential distribution process, the following description is based on the strong trust assumption that the authorities and the actions invoked within the untappable channel are done without any adversarial context.

[3] A more detailed description of the physical and chemical properties of secure printing can be found in the dissertation of Aleksander Essex.

[4] This value is arbitrarily chosen and represents a possible – usable – value

'package' of τ-credentials such that the trusted voting device is in possession of all presented credentials. Please note, that the amount of τ-'packages' per voter is not restricted in theory but will require an upper limit in practice.

Multi-Encryption Publication: The voting device then stores the credentials within a multi-encryption ciphertext, which can be published on the public bulletin board. This step allows to hand over the management of the multi-encryption ciphertext to the system, releasing the human voter as well as the trusted device from this task.

After this process, the trusted voting device clears all internal memory of the credentials and the now registered voter can leave the registration office (untappable channel) without any physical traces of the registration process, except maybe a cleared trusted voting device. If the human voter should ever be left without a working trusted voting device, it is always possible to 'grab' a new one at any time at the registration office.

10.3. Voting Event Preparation

The election authorities provide the voting material. It includes information about the voting event, the 'question', the selectable options and the selection policy. This data has to be provided in such a way that the human voter as well as the trusted device are able to read it. Finally the data is signed so that it can be verified by the trusted device using sk_E, and is then published on the public bulletin board.

10.4. Vote Casting

The human voter uses the personal computer in order to reach and manage the voting material from the public bulletin board. The final selection of the volition is then made by the human voter by reading the actual selection into the trusted voting device.

The device then verifies if the selection made is valid in terms of the given voting policies and if it carries a valid signature, verifiable with sk_E. Only if these tests succeed will the trusted voting device proceed in presenting the selection to the voter. In any other case, the voter will be informed about the inconsistency so that they can act accordingly.

Vote Encryption: Only if the voter accepts the selection presented on the secure display, the device encrypts the vote using the public key pk_T together with the required zero knowledge proofs.

Credential Encryption: Finally the voter is requested to read in the personal multi-encryption ciphertext version (from the PBB) to the trusted voting device and enters the corresponding password (for either the true credential σ or a fake credential τ). The trusted voting device then uses the retrieved credential and encrypts it, this time by using the public key pk_C, together with a zero knowledge proof of the plaintext. The accumulated and encrypted data record is then stored as a file on the trusted voting device, ready to be read out and sent to the public bulletin board.

10.5. Vote Processing

Once the data record, containing the pk_C encrypted credential and the pk_T encrypted vote has reached the public bulletin board (via an anonymous channel), the set of credential authorities CA verifies if the encrypted credential is valid (represents either a τ or a σ) and whether this credential is not yet present at the public bulletin board.

Invalid Credential: If the credential is not valid, the record is publicly labeled as invalid and rejected by the public bulletin board. This way, the voter can be informed about the reject and can act accordingly (i.e. by repeating the credential encryption process).

Duplicate Credential: If the credential is already present on the public bulletin board, the record will be labeled accordingly and rejected publicly. This gives the voter special information. If, for example, no vote whatsoever has been placed by this specific voter, very strong evidence is available that the e-voting system has been attacked (successfully) via this credential. The same is true if a voter succeeds in casting a vote but realizes that some 'foreign' cast will be recognized as duplicate. This provides strong evidence that the e-voting system has been attacked (without success this time) via this credential. The voter reaction may vary at this time, ranging from simply choosing another credential to re-registering for the next voting period.[5]

If neither of the described cases lead to a reject, the vote is accepted to be stored in the e-voting system on the public bulletin board. Acceptance of the vote provides strong individual evidence that the vote indeed has been cast as intended.

[5] In the worst case, the adversary managed to post a on the public bulletin board using the voter's σ credential. Even though the voter will realize that immediately when trying to post the true vote, by design, the voter is not able to prove the attack to others. If there is no organizational way to override a remotely posted vote (e.g., by voting in person at the polling station), the voter has lost the right to vote for this voting event.

10.6. Verification

After a successful vote cast, voters can individually verify whether their vote has been stored correctly on the public bulletin board. As a human voter is not capable of making the calculations, the trusted voting device takes over this part as well.

For individual verifiability, the public bulletin board has to give a representation of all the encrypted votes readable for humans and for the trusted voting device. The device then compares the encrypted votes on the public bulletin board with the encrypted vote created on the voter's side. If they show the same shape (bitwise), the trusted voting device confirms it to the human voter. Due to the zero knowledge proofs required from all processes for mixing, re-encrypting, plaintext equivalence tests and decrypting, these steps can all be verified universally. This way, no process involved can fail in its duty without being noticed and E2E-verifiability is achieved. However, these verifications require a lot of computing power. So, these tests might be left for the already introduced *layer of experts*. Please note, that everyone can become part of this layer at any time, so there is no dedicated organization or limitation of this predicate. If at least one of the experts finds and presents a failing proof, this fact can easily be confirmed by the remaining group members. Unfortunately though, the opposite is not true and hence every single member of the group of experts has to verify the complete set of proofs if no proof fails. These verifications result in trust for correct functioning of the remote e-voting system, as long as there are enough independent experts involved in the process.

10.7. Summary

A coercion-resistant E2E-verifiable remote e-voting system with usability aspects is feasible at last. However, the demonstration given also hints on the shier complexity of it, requiring special hardware, on the voter's side as well as at the 'server' side during the setup of the e-voting system.

The demonstration given is but a vague sketch of a real instantiation and will probably need some more dissertations exploring usability and implementation issues within this field.

Chapter 11

Conclusion

This dissertation presents contributions within the field of coercion-resistant E2E-verifiable remote e-voting leading it towards reality. By a minimal relaxation of the original voter model towards the human voter, the limitations in the usability of the original JCJ05-scheme and of most of its derivatives are demonstrated. This dissertation presents the only known JCJ05-derivative capable of coping with the requirements of the relaxed voter model. In addition, this JCJ05-derivative overcomes two major problems of the original JCJ05-scheme, namely the quadratic tallying time and the possibility of an extrinsically provoked denial of service at the protocol-level. The added complexity of the credential handling on the voter's side is compensated by the introduction of a new cryptographic component capable of handling credentials on behalf of the human voter. Finally, a solution to the secure platform problem is presented by introducing a trusted voting device with limited computational capabilities, defining a true interface between the voter's brain and the voter's untrustworthy computer equipment preserving the secrecy of the vote on the voter's side.

However, this thesis does not deal with the problem of everlasting privacy, but rather shows that this remains an organizational problem of access control and is ultimately based on strong trust assumptions.

"It is as if we were extracting bullets from a loaded revolver" Returning to the metaphorical citation used as an introduction to this dissertation, this work presents the extraction of several bullets, such as the secure platform problem, the denial of service at protocol level and the coercion-resistance comprising usability aspects for the human voter, but it is not able to unload the weapon completely, as the bullet of (everlasting) privacy remains stuck.

References

[1] M. Abadi, T. Mark, A. Lomas, and Roger Needham. Strengthening passwords. Technical report, SRC Technical Note, 1997.

[2] B. Adida. Helios: Web-based open-audit voting. In P. Van Oorschot, editor, *SS'08, 17th USENIX Security Symposium*, pages 335–348, San Jose, USA, 2008.

[3] R. Araújo, S. Foulle, and J. Traoré. A practical and secure coercion-resistant scheme for remote elections. In D. Chaum, M. Kutylowski, R. L. Rivest, and P. Y. A. Ryan, editors, *FEE'07, Frontiers of Electronic Voting*, pages 330–342, Schloss Dagstuhl, Germany, 2007.

[4] R. Araújo, R. Robbana N. Ben Rajeb, J. Traoré, and S. Youssfi. Towards practical and secure coercion-resistant electronic elections. In S. H. Heng, R. N. Wright, and B. M. Goi, editors, *CANS'10, 9th International Conference on Cryptology And Network Security*, LNCS 6467, pages 278–297, Kuala Lumpur, Malaysia, 2010.

[5] Isaac Asimov and James L. Quinn. *if, Worlds of Science Fiction*, volume 5. Quinn Publishing Company, Inc.; Buffalo, NY, August 1955.

[6] E. Bangerter, J. Camenisch, and U. Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In S. Vaudenay, editor, *PKC'05, 8th International Workshop on Theory and Practice in Public Key Cryptography*, LNCS 3386, pages 154–171, Les Diablerets, Switzerland, 2005.

[7] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. In *In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393. IEEE Computer Society, 2004.

[8] Gregory V. Bard. Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*, ACSW '07, pages 117–124, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[9] J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections. In *STOC'94, 26th Annual ACM Symposium on Theory of Computing*, pages 544–553, Montréal, Canada, 1994.

[10] J. D. C. Benaloh. *Verifiable Secret-Ballot Elections.* PhD thesis, Yale University, New Haven, USA, 1987.

[11] D. Berger and R. Linder. Sicheres und effizientes e-voting. Project report, Bern University of Applied Sciences, Biel, Switzerland, 2011.

[12] D. Bernhard, O. Pereira, and B. Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012.

[13] J. Beuchart. Append-only web bulletin board. Project report, Bern University of Applied Sciences, Biel, Switzerland, 2011.

[14] Eric A. Brewer. Towards robust distributed systems. In *Symposium on Principles of Distributed Computing (PODC)*, 2000.

[15] A. S. Brown, E. Bracken, S. Zoccoli, and K. Douglas. Generating and remembering passwords. *Applied Cognitive Psychology*, 18(6):641–651, 2004.

[16] 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 Melanie Volkamer Manuel J. Kripp and Rüdiger Grimm, editors, *5th International Conference on Electronic Voting 2012 (EVOTE2012)*, 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, Jul 2012.

[17] Matt Burnside, Dwaine Clarke, Blaise Gassend, Thomas Kotwal, Marten van Dijk, Srinivas Devadas, and Ronald [L.] Rivest. The untrusted computer problem and camera-based authentication. In Friedemann Mattern and Mahmoud Naghshineh, editors, *Proceedings of the International Conference*

on Pervasive Computing, volume 2414 of *Lecture Notes in Computer Science*, pages 114–124. Springer.

[18] Sergiu Bursuc, Gurchetan S. Grewal, and Mark Dermot Ryan. Trivitas: Voters directly verifying votes. In Aggelos Kiayias and Helger Lipmaa, editors, *VOTE-ID*, volume 7187 of *Lecture Notes in Computer Science*, pages 190–207. Springer, 2011.

[19] Jon Callas, Chief Technology Officer, and Chief Security Officer. *An Introduction to Cryptography*. PGP Corporation, 2006.

[20] J. Camenisch, J. M. Piveteau, and M. Stadler. Blind signatures based on the discrete logarithm problem. In A. De Santis, editor, *EUROCRYPT'94, International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 950, pages 428–432, Perugia, Italy, 1994.

[21] Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. Power of randomization in automata on infinite strings. In *Proceedings of the 20th International Conference on Concurrency Theory*, CONCUR 2009, pages 229–243, Berlin, Heidelberg, 2009. Springer-Verlag.

[22] D. Chaum. Blind signature system. In *CRYPTO'83, 3rd International Cryptology Conference*, pages 153–156, Santa Barbara, USA, 1983.

[23] D Chaum. Blind Signature System. In *{CRYPTO'83}, 3rd International Cryptology Conference*, pages 153–156, Santa Barbara, USA, 1983.

[24] D. Chaum, A. Essex, R. Carback, J. Clark, S. Popoveniuc, A. Sherman, and P. Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy*, 6(3):40–46, 2008.

[25] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981.

[26] J. Clark and U. Hengartner. Selections: Internet voting with over-the-shoulder coercion-resistance. In *FC'11, 15th International Conference on Financial Cryptography*, St. Lucia, 2011.

[27] Jeremy Clark and Urs Hengartner. Panic passwords: authenticating under duress. In *Proceedings of the 3rd conference on Hot topics in security*, HOTSEC'08, pages 8:1–8:6, Berkeley, CA, USA, 2008. USENIX Association.

[28] M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. Technical Report TR 2007-2081, Department of Computer Science, Cornell University, 2007.

[29] M. R. Clarkson, S. Chong, and A. C. Myers. Civitas: Toward a secure voting system. In *SP'08, 29th IEEE Symposium on Security and Privacy*, pages 354–368, Oakland, USA, 2008.

[30] Scott Contini, Ron Steinfeld, Josef Pieprzyk, and Krystian Matusiewicz. A critical look at cryptographic hash function literature. *Coding and Cryptology - Proceedings of the First International Workshop*, pages 58–79, 2008.

[31] Véronique Cortier and Ben Smyth. Attacking and fixing helios: An analysis of ballot secrecy. *Journal of Computer Security*, 2012. to appear.

[32] R. Cramer, I. Damgård, and J. B. Nielsen. Multiparty computation, an introduction. Lecture notes, Department of Computer Science, University of Aarhus, Denmark, 2009.

[33] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *EUROCRYPT'97, International Conference on the Theory and Application of Cryptographic Techniques*, LNCS 1233, pages 103–118, Konstanz, Germany, 1997.

[34] R.J.F. Cramer. *Modular Design of Secure Yet Practical Cryptographic Protocols*. 1997.

[35] Department of Defense. *Trusted Computer System Evaluation Criteria*. Department of Defense, 1985.

[36] R. Di Cosmo. On privacy and anonymity in electronic and non electronic voting: the ballot-as-signature attack. *Hyper Articles en Ligne*, hal-00142440(2), 2007.

[37] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In M. Blaze, editor, *SS'04, 13th USENIX Security Symposium*, pages 303–320, San Diego, USA, 2004.

[38] E. Dubuis, R. Haenni, and R. E. Koenig. Konzept und Implikationen eines verifizierbaren Vote Électronique Systems. Studie im Aufrag der Schweizerischen Bundeskanzlei, April 2012.

[39] Donald E. Eastlake, Jeffrey I. Schiller, and Steve Crocker. Randomness Requirements for Security. In *BCP 106, RFC 4086*, 2005.

[40] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84, Advances in Cryptology*, LNCS 196, pages 10–18, Santa Barbara, USA, 1984. Springer.

[41] Aleksander Essex, Jeremy Clark, and Urs Hengartner. Cobra: toward concurrent ballot authorization for internet voting. In *Proceedings of the 2012 international conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'12, pages 3–3, Berkeley, CA, USA, 2012. USENIX Association.

[42] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86, 6th Annual International Cryptology Conference on Advances in Cryptology*, pages 186–194, Santa Barbara, USA, 1986.

[43] D. Florêncio and C. Herley. A Large-Scale Study of Web Password Habits. In P. F. Patel-Schneider and P. Shenoy, editors, *WWW'07, 16th International Conference on World Wide Web*, pages 657–666, Banff, Canada, 2007.

[44] D. Florêncio and C. Herley. A large-scale study of web password habits. In P. F. Patel-Schneider and P. Shenoy, editors, *WWW'07, 16th International Conference on World Wide Web*, pages 657–666, Banff, Canada, 2007.

[45] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, ASIACRYPT '92, pages 244–251, London, UK, 1993. Springer-Verlag.

[46] S. Gaw and E. W. Felten. Password Management Strategies for Online Accounts. In L. F. Cranor, editor, *SOUPS'06, 2nd Symposium on Usable Privacy and Security*, pages 44–55, Pittsburgh, USA, 2006.

[47] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC'09, 41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, USA, 2009.

[48] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent available partition-tolerant web services. In *In ACM SIGACT News*, page 2002, 2002.

[49] I. Goldberg. On the security of the Tor authentication protocol. In G. Danezis and P. Golle, editors, *PET'06, 6th Workshop on Privacy Enhancing Technologies*, pages 316–331, Cambridge, U.K., 2006.

[50] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[51] Oded Goldreich. On expected probabilistic polynomial-time adversaries: A suggestion for restricted definitions and their benefits. *Journal of Cryptology*, 23:1–36, 2010. 10.1007/s00145-009-9050-5.

[52] J. Groth. Short non-interactive zero-knowledge proofs. In M. Abe, editor, *ASIACRYPT'10, 16th International Conference on the Theory and Application of Cryptographic Techniques*, LNCS 6477, pages 341–358, Singapore, 2010.

[53] R. Haenni, E. Dubuis, and R. E. Koenig. Konzept und Implikationen eines verifizierbaren Vote Electronique Systems. Federal Chancellery Switzerland, 2012. `http://www.bk.admin.ch/themen/pore/evoting/07977/index.html?download=M3wBPgDB_8ul16Du36WenojQ1NTTjaXZnqWfVpzLhmfhnapmmc7Zi6rZnqCkkId3f359bKbXrZ6lhuDZz8mMps2g`

[54] R. Haenni, R. E. Koenig, and E. Dubuis. Voting over the Internet on an Insecure Platform. In D. Zissis and D. Lekkas, editors, *Design, Development, and Use of Secure Electronic Voting Systems*, page To be published. IGI Global, 2013.

[55] R. Haenni and O. Spycher. Secure internet voting on limited devices with anonymized DSA public keys. In *EVT/WOTE'11, Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, San Francisco, USA, 2011.

[56] Rolf Haenni and Reto E. Koenig. A Generic Approach to Prevent Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes. *Computers & Security*, 2013.

[57] J. Heather and D. Lundin. The append-only web bulletin board. In P. Degano, J. Guttman, and F. Martinelli, editors, *FAST'08, 5th International Workshop on Formal Aspects in Security and Trust*, LNCS 5491, pages 242–256, Malaga, Spain, 2008.

[58] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *EUROCRYPT'00, International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 1807, pages 539–556, Bruges, Belgium, 2000.

[59] Jaydeep Howlader, Vivek Nair, Saikat Basu, and AK Mal. Uncoercibility in e-voting and e-auctioning mechanisms using deniable encryption. *International Journal of Network Security Its Applications IJSNA*, 3:97–107, 2011.

[60] M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. In T. Okamoto, editor, *ASIACRYPT'00, 6th International Conference on the Theory and Application of Cryptographic Techniques*, LNCS 1976, pages 162–177, Kyoto, Japan, 2000.

[61] M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In D. Boneh, editor, *SS'02, 11th USENIX Security Symposium*, pages 339–353, San Francisco, USA, 2002.

[62] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In U. Maurer, editor, *EUROCRYPT'96, International Conference on the Theory and Application of Cryptographic Techniques*, LNCS 1070, pages 143–154, Saragossa, Spain, 1996.

[63] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In V. Atluri, S. De Capitani di Vimercati, and R. Dingledine, editors, *WPES'05, 4th ACM Workshop on Privacy in the Electronic Society*, pages 61–70, Alexandria, USA, 2005.

[64] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[65] John Kelsey, Bruce Schneier, Chris Hall, and David Wagner. Secure applications of low-entropy keys. In *LECTURE NOTES IN COMPUTER SCIENCE*, pages 121–134. Springer-Verlag, 1998.

[66] J. Kim, K. Kim, and C. Lee. An efficient and provably secure threshold blind signature. In K. Kim, editor, *ICISC'01, 4th International Conference on Information Security and Cryptology*, LNCS 2288, pages 318–327, Seoul, South Korea, 2002.

[67] R. Koenig. How to store some secrets. In R. M. Alvarez, J. Benaloh, A. Rosen, and P. Y. A. Ryan, editors, *Dagstuhl Seminar Nr. 11281: Verifiable Elections and the Public*, number 7 in Dagstuhl Reports, pages 45–45, Dagstuhl, Germany, 2011.

[68] R. Koenig, E. Dubuis, and R. Haenni. Why public registration boards are required in e-voting systems based on threshold blind signature protocols. In R. Krimmer and R. Grimm, editors, *EVOTE'10, 4th International Workshop on Electronic Voting*, number P-167 in Lecture Notes in Informatics, pages 255–266, Bregenz, Austria, 2010. Gesellschaft für Informatik E.V.

[69] R. Koenig, R. Haenni, and S. Fischli. Preventing board flooding attacks in coercion-resistant electronic voting schemes. In J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, and C. Rieder, editors, *SEC'11, 26th IFIP International Information Security Conference*, volume 354, pages 116–127, Lucerne, Switzerland, 2011.

[70] R. E. Koenig and R. Haenni. How to Store some Secrets. Cryptology ePrint Archive, Report 2012/375, 2012. `http://eprint.iacr.org/`.

[71] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2595–2604, New York, NY, USA, 2011. ACM.

[72] Stephan Krenn. *Bringing Zero-Knowledge Proofs of Knowledge to Practice*. PhD thesis, University of Fribourg (Switzerland), 2012.

[73] R. Küsters, T. Truderung, and A. Vogt. A game-based definition of coercion-resistance and its applications. In A. Myers and M. Backes, editors, *CSF'10, 23nd IEEE Computer Security Foundations Symposium*, pages 122–136, Edinburgh, U.K., 2010.

[74] B. Lee and K. Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In P. J. Lee and C. H. Lim, editors, *ICISC'02, 5th International Conference on Information Security and Cryptology*, LNCS 2587, pages 389–406, Seoul, South Korea, 2002.

[75] S. Lee, R. Sherwood, and S. Bhattacharjee. Cooperative peer groups in NICE. In *IEEE-INFOCOM'03, 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1272–1282, San Francisco, USA, 2003.

[76] Arjen K. Lenstra. Key Length, 2004.

[77] J. T. Liechti and L. Bernath. KryptonIT – Password Tresor. Bachelor thesis, Bern University of Applied Sciences, Biel, Switzerland, 2012.

[78] E. Magkos, M. Burmester, and V. Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In B. Schmid, K. Stanoevska-Slabeva, and V. Tschammer, editors, *I3E'01, 1st IFIP Conference on towards the E-Society*, volume 202, pages 683–694, 2001.

[79] Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In *AFRICACRYPT*, pages 272–286, 2009.

[80] G. Meister, D. Hühnlein, J. Eichholz, and R. Araujo. eVoting with the European citizen card. In A. Brömme, C. Busch, and D. Hühnlein, editors, *BIOSIG'08, Special Interest Group on Biometrics and Electronic Signatures*, number P-137 in Lecture Notes in Informatics, pages 67–78, Darmstadt, Germany, 2008. Gesellschaft für Informatik E.V.

[81] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, USA, 1996.

[82] R. Mercuri. A better ballot box? *IEEE Spectrum*, 39(10):46–50, 2002.

[83] T. Moran and M. Naor. Receipt-free universally-verifiable voting with everlasting privacy. In C. Dwork, editor, *CRYPTO'06, 26th Annual International Cryptology Conference on Advances in Cryptology*, LNCS 4117, pages 373–392, Santa Barbara, USA, 2006.

[84] C. A. Neff. Practical high certainty intent verification for encrypted votes. Technical report, VoteHere, Inc., 2004.

[85] Brett Ninness. Strong Laws of Large Numbers Under Weak Assumptions with Application. *IEEE TRANSACTIONS ON*, 45:2000, 1999.

[86] H. Nyquist. Thermal Agitation of Electric Charge in Conductors. *Phys. Rev.*, 32:110–113, Jul 1928.

[87] R. Oppliger. How to address the secure platform problem for remote internet voting. In *SIS'02, 5th Conference on "Sicherheit in Informationssystemen"*, pages 153–173, Vienna, Austria, 2002.

[88] A. Pellegrini and P. von Bergen. SwissiVi. Project report, Bern University of Applied Sciences, Biel, Switzerland, 2012.

[89] B. Pfitzmann. Breaking an efficient anonymous channel. In A. De Santis, editor, *EUROCRYPT'94, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 950 of *LNCS 950*, pages 332–340, Perugia, Italy, 1995.

[90] S. Popoveniuc and E. Leontie. Safe RPC: Auditing mixnets safely using randomized partial checking. In P. Samarati and S. Katsikas, editors, *SECRYPT'10, 5th International Conference on Security and Cryptography*, Athens, Greece, 2010.

[91] J. J. Quisquater, L. Guillou, and T. Berson. How to explain zero-knowledge protocols to your children. In G. Brassard, editor, *CRYPTO'89, 9th Annual International Cryptology Conference on Advances in Cryptology*, LNCS 435, pages 628–631, Santa Barbara, USA, 1989.

[92] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[93] R. L. Rivest, A. Shamir, and L. M. Adelman. A method for obtaining digital signatures and public-key cryptosystems. Technical Report TM-82, MIT, Cambridge, USA, 1977.

[94] Ronald L. Rivest. Electronic voting. Talk given for Cambridge Club at Harvard Faculty Club, March 5, 2001.

[95] P. Y. A. Ryan. Prêt à voter with confirmation codes. In *EVT/WOTE'11, Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, San Francisco, USA, 2011.

[96] P. Y. A. Ryan, D. Bismark, J. Heather, S. Schneider, and X. Zhe. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4):662–673, 2009.

[97] K. Sako and J. Kilian. Receipt-free mix-type voting scheme: A practical solution to the implementation of a voting booth. In L. C. Guillou and J. J. Quisquater, editors, *EUROCRYPT'95, 15th International Conference on the Theory and Applications of Cryptographic Techniques*, LNCS 921, pages 393–403, Saint-Malo, France, 1995.

[98] K. Sampigethaya and R. Poovendran. A survey on mix networks and their secure applications. *IEEE*, 94(12):2142–2181, 2006.

[99] M. Schläpfer, R. Haenni, R. Koenig, and O. Spycher. Efficient vote authorization in coercion-resistant internet voting. In *VoteID'11, 3rd International Conference on E-Voting and Identity*, Tallinn, Estonia, 2011.

[100] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[101] J. Schweisgut. Coercion-resistant electronic elections with observer. In R. Krimmer, editor, *EVOTE'06, 2nd International Workshop on Electronic Voting*, number P-86 in Lecture Notes in Informatics, pages 171–177, Bregenz, Austria, 2006. Gesellschaft für Informatik E.V.

[102] J. Schweisgut. Effiziente elektronische wahlen mit observer. In J. Dittmann, editor, *Sicherheit 2006*, number P-77 in Lecture Notes in Informatics, pages 306–316, Magdeburg, Germany, 2006. Gesellschaft für Informatik E.V.

[103] J. Schweisgut. *Elektronische Wahlen unter dem Einsatz kryptografischer Observer*. PhD thesis, Fachbereich Mathematik und Informatik, Physik, Geographie Justus-Liebig-Universität Gießen, Deutschland, 2007.

[104] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[105] W. D. Smith. New cryptographic voting scheme with best-known theoretical properties. In *FEE'05, Workshop on Frontiers in Electronic Elections*, Milan, Italy, 2005.

[106] O. Spycher and R. Haenni. A novel protocol to allow revocation of votes in a hybrid voting system. In *ISSA'10, 9th Annual Conference on Information Security – South Africa*, Sandton, South Africa, 2010.

[107] O. Spycher, R. Koenig, R. Haenni, and M. Schlaepfer. Achieving meaningful efficiency in coercion-resistant, verifiable internet voting. In M. Kripp, editor, *EVOTE'12, 5th International Workshop on Electronic Voting*, Lecture Notes in Informatics, Bregenz, Austria, 2012 (submitted). Gesellschaft für Informatik E.V.

[108] O. Spycher, R. Koenig, R. Haenni, and M. Schläpfer. A new approach towards coercion-resistant remote e-voting in linear time. In *FC'11, 15th International Conference on Financial Cryptography*, St. Lucia, 2011.

[109] O. Spycher and M. Volkamer. Measures to establish trust in Internet voting. In *ICEGOV'11, 5th International Conference on Theory and Practice of Electronic Governance*, Tallinn, Estonia, 2011.

[110] O. Spycher, M. Volkamer, and R. Koenig. Transparency and technical measures to establish trust in Norwegian Internet voting. In *VoteID'11, 3rd International Conference on E-Voting and Identity*, Tallinn, Estonia, 2011.

[111] F. Stajano. Pico: No more passwords! In *IWSP'11, 19th Security Protocols Workshop*, Cambridge, U.K., 2011.

[112] Processing Standards. Secure hash standard (shs). *Processing*, FIPS PUB 1(October), 2008.

[113] G. Weber, R. Araujo, and J. Buchmann. On coercion-resistant electronic elections with linear work. In *ARES'07, 2nd International Conference on Availability, Reliability and Security*, pages 908–916, Vienna, Austria, 2007.

[114] S. Weber. *Coercion-Resistant Cryptographic Voting: Implementing Free and Secret Electronic Elections*. VDM Verlag, Saarbrücken, Germany, 2008.

[115] Thomas Weigold, Thorsten Kramp, Reto Hermann, Frank Höring, Peter Buhler, and Michael Baentsch. The zurich trusted information channel — an efficient defence against man-in-the-middle and malicious software attacks. In *Proceedings of the 1st international conference on Trusted Computing and Trust in Information Technologies: Trusted Computing - Challenges and Applications*, Trust '08, pages 75–91, Berlin, Heidelberg, 2008. Springer-Verlag.

[116] D. Wikström. A commitment-consistent proof of a shuffle. In C. Boyd and J. González Nieto, editors, *ACISP'09, 14th Australasian Conference on Information Security and Privacy*, LNCS 5594, pages 407–421, Brisbane, Australia, 2009.

[117] A. Zúquete. Enhanced secure interface for a portable e-voting terminal. In E. Fernández-Medina, M. Malek, and J. Hernando, editors, *SECRYPT'08, 3rd International Conference on Security and Cryptography*, pages 529–537, Porto, Portugal, 2008.

[118] A. Zúquete, C. Costa, and M. Romão. An intrusion-tolerant e-voting client system. In M. Correia and N. Ferreira Neves, editors, *WRAITS'07, 1st Workshop on Recent Advances on Intrusion-Tolerant Systems*, pages 23–27, Lisbon, Portugal, 2007.

Curriculum Vitae

1973	Born on March 1^{st} in Basel, Switzerland
1980 – 1989	Primary- and secondary school in Therwil, Switzerland
1993 – 1997	Study (HTL) computer science at Höhere Technische Lehranstalt Biel, Switzerland
2005 – 2007	Study (MSc.) computer science at University of Fribourg, Switzerland
2009 – 2013	PhD-student at University of Fribourg, Switzerland
since 2009	Full professor in computer science at Bern University of Applied Sciences

Ehrenwörtliche Erklärung

Hiermit bestätige ich mit meiner Unterschrift, dass ich die hier vorgelegte Dissertation persönlich verfasst und dabei nur die angeführten Quellen und Hilfsmittel verwendet habe; wörtliche Zitate und Paraphrasen sind als solche gekennzeichnet.

Ich habe zur Kenntnis genommen, dass wissenschaftliches Fehlverhalten nach den Richtlinien der Universität Freiburg[1] geahndet wird.

Titel der Arbeit:
Electronic Voting over the Internet – The Boon and Bane of Modern E-Society

Vorname:
Reto Eric

Name:
Koenig

Ort und Datum:
Fribourg, 15. Oktober 2013

Unterschrift:

[1] Richtlinien der Universität Freiburg vom 13. Mai 2008 über das Verfahren für die Verhängung von Disziplinarstrafen nach Art. 101 der Statuten der Universität Freiburg vom 31. März 2000 im Falle des Verstoßes gegen die Regeln guter wissenschaftlicher Praxis beim Verfassen schriftlicher Arbeiten während der Ausbildung: `http://www.unifr.ch/rectorat/reglements/pdf/1_1_15.pdf` Art. 2: „Wissenschaftliches Fehlverhalten liegt vor, wenn gegen die Regeln guter wissenschaftlicher Praxis verstoßen wird, namentlich wenn in einer schriftlichen Arbeit fremde Arbeitsergebnisse und Erkenntnisse unter eigenem Namen verfasst werden (Plagiat), wenn eine Arbeit eingereicht wird, die von einer Drittperson verfasst worden ist (Ghostwriting), oder wenn vorsätzlich oder grob fahrlässig Falschangaben gemacht werden“.