Department of Informatics
University of Fribourg (Switzerland)

Trustworthy Internet Voting

-

Defeating Powerful Coercers and Vote-Buyers

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

Oliver Spycher

from

Bern, Switzerland

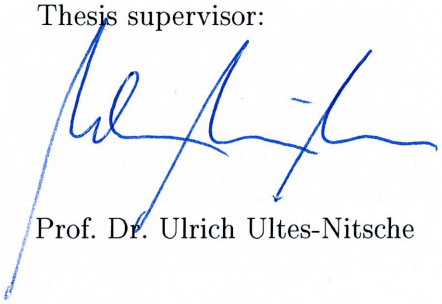
Thesis No. 1922
UniPrint, Fribourg
2015

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. Melanie Volkamer, Technische Universität Darmstadt,
- Prof. Dr. Eric Dubuis, Bern University of Applied Sciences.

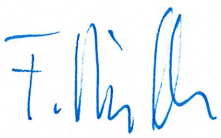
Fribourg, August 31, 2015

Thesis supervisor:



Prof. Dr. Ulrich Ultes-Nitsche

Dean:



Prof. Dr. Fritz Müller

Acknowledgement

I warmly thank my family, friends, advisors and colleagues for their care and guidance.

This research has been funded by the Hasler Foundation.

Abstract

Votes need to be validated and processed without breaking the voters' privacy. There are solutions to these opposing requirements that can be implemented without introducing any single point of failure. Particularly, even in the event of a few corrupted components, the votes can still be validated and processed correctly. Due to a strong sense of separation of duty, it would take a whole set of corrupted system components to aggregate enough information to break the secrecy of the ballot. Verifiability takes this even a step further. With verifiable systems anyone could be invited to verify that the votes have been handled correctly. Particularly, voters can be allowed to verify that their own votes have been considered in the final tally unchanged. This can be achieved by furnishing zero-knowledge proofs that are highly sound. Even verifiability can be granted without breaking the voters' privacy. Verifiability and separation of duty are the ingredients to convince that Internet voting is trustworthy. The thesis introduces cryptographic building-blocks suitable for defining such systems.

Coercion-resistance can be considered a stronger form of privacy, allowing voters who are put under pressure yet to vote freely. A system that offers coercion-resistance allows a voter to pretend having followed a coercer's instructions even when he has not. Although it may seem that verifiability and coercion-resistance are mutually exclusive, in 2005 Juels, Catalano and Jakobsson have come up with a promising approach. Yet, some issues are left open. Particularly, the tallying time is proportional to the square of the number of voters, thus yielding tallying unbearably long in large-scale votes. This thesis introduces two solutions that drastically reduce the time at tallying. Both schemes are observed in relation with other ones known from the literature.

Conventional means of voting are unlikely to be replaced by Internet voting within the next decades. The question arises, how the Internet channel can be integrated appropriately. The thesis proposes hybrid schemes to allow voters to revoke their vote cast through the Internet by using a physical ballot, thus yielding the overall system coercion-resistant. It also proposes two different procedures for revoking votes and specifies the requirements an Internet voting protocol needs to meet in such a setting. It proposes two protocols that meet these requirements and seem to be good choices for the Internet channel. As a special feature, they offer verifiability, without allowing to find out whether voters have participated at all. They are much simpler to put into practice than the ones mentioned above. One of the protocols has been used in the *Selectio Helvetica* system, which has hosted votes from the *Baloti* project. In the meantime the protocol has been implemented within *UniVote* and has been used to host several student board elections of Swiss universities.

Zusammenfassung

Die Gültigkeitsprüfung und die Verarbeitung von Stimmen muss unter Berücksichtigung des Stimmgeheimnisses erfolgen. Für diese scheinbar widersprüchlichen Anforderungen gibt es Lösungen, die Klumpenrisiken gänzlich umgehen. Namentlich können die Gültigkeitsprüfung und die Verarbeitung auch dann korrekt erfolgen, wenn einige der Systemkompomenten manipuliert worden sind. Dank einer rigiden Arbeitsteilung ist sichergestellt, dass keine Komponente über genügend Informationen verfügt, um das Stimmgeheimnis zu brechen. Dazu müssten mehrere Komponenten manipuliert werden. Die Verfizierbarkeit geht noch einen Schritt weiter. Namentlich können die einzelnen Stimmenden mit einem verifizierbaren System überprüfen, ob ihre Stimme in unveränderter Form im Ergebnis berücksichtigt ist. Dies kann auf der Grundlage von sog. Zero-Knowledge Beweisen erfolgen, die äusserst zuverlässig sind. Sogar die Verifizierbarkeit kann unter voller Berücksichtigung des Stimmgeheimnisses gewährleistet werden. Die Verifizierbarkeit sowie die Aufteilung von Verantwortlichkeiten bilden die Zutaten, auf deren Grundlage sich die Vertrauenswürdigkeit von E-Voting über das Internet auf überzeugende Weise erklären lässt. Die vorliegende Dissertation bietet eine Einführung in einige kryptografische Grundkomponenten, die die Entwicklung solcher Systeme ermöglichen.

Der Schutz gegen Erpressung (engl. coercion-resistance) ist eine starke Form des Stimmgeheimnisses. Sogar Stimmende, die unter Druck gesetzt werden, können damit ihre Stimme weiterhin frei abgegeben. Ein System, das Schutz gegen Erpressung bietet, erlaubt es den Stimmenden, gegenüber beliebigen Akteuren ein beliebiges Stimmverhalten vorzutäuschen. Auf den ersten Blick mag es scheinen, dass sich der Schutz gegen Erpressung nicht mit der Verfizierbarkeit vereinbaren lässt. Dennoch ist Juels, Catalano und Jakobsson im Jahr 2005 ein spannender Lösungsansatz geglückt. Einige Schwierigkeiten blieben allerdings bestehen. Namentlich steigt die Wartezeit bei der Auszählung der Stimmen quadratisch zur Anzahl Stimmberechtigter. Für gross angelegte Wahlen und Abstimmungen ist die vorgeschlagene Lösung damit ungeeignet. Die vorliegende Dissertation stellt zwei Lösungen vor, die eine drastische Kürzung der Wartezeit bei der Auszählung ermöglicht. Die beiden Lösungen werden mit anderen Lösungen in der Literatur verglichen.

Die konventionellen Arten der Stimmabgabe werden in den nächsten Jahren kaum gänzlich mit der Stimmabgabe über das Internet ersetzt werden. Dies führt unweigerlich zur Frage, wie die Stimmabgabe über das Internet in die bereits bestehenden Prozesse eingebettet werden soll. Die Dissertation stellt dazu sog. hybride Systeme vor. Hybride

Syteme erlauben es den Stimmenden, eine über das Internet abgegebene Stimme zu ersetzen, indem sie eine zweite Stimme über einen konventionellen Kanal abgeben. Das Gesamtsystem bietet damit einen Schutz gegen Erpressung, die auf der Grundlage des konventionellen Stimmkanals erfolgt. Weiter werden zum Ersetzen der ersten Stimme zwei grundlegende Umsetzungsmöglichkeiten und die jeweiligen Anforderungen an das kryptografische Protokoll für den Internetkanal vorgestellt. Die Dissertation beschreibt zwei Protokolle, die diese Anforderungen erfüllen und sich als Lösungen für den Internetkanal eignen. Sie zeichnen sich dadurch aus, dass sie die Verifizierbarkeit bieten und gleichzeitig keine Informationen erheben, welche Stimmberchtigten tatsächlich eine Stimme abgegeben haben. Trotz dieser Eigenschaft sind sie deutlich leichter umzusetzen, als jene, die losgelöst von der konventionellen Stimmabgabe funktionieren. Eines der beiden Protokolle kam im System Selectio Helvetica zur Anwendung. Das System hat beim Baloti-Projekt die elektronische Stimmabgabe ermöglicht. In der Zwischenzeit verwendet das System UniVote das Protokoll. Das System kam bereits bei mehreren Studentenratswahlen zum Einsatz.

Contents

1 Introduction **1**
- 1.1 Contents of this Thesis . . . 2
- 1.2 Contribution . . . 3

2 Common Building Blocks to Allow Trustworthy Internet Voting **5**
- 2.1 Definitions . . . 5
- 2.2 Security Requirements . . . 6
- 2.3 Trust Assumptions . . . 10
- 2.4 Public Bulletin Boards . . . 12
- 2.5 Homomorphic Randomized Encryption and ElGamal Cryptosystem . . . 13
- 2.6 Non-Interactive Zero-Knowledge Proofs . . . 15
- 2.7 Secure Multiparty Computation . . . 18
- 2.8 Verifiable Mix-Nets . . . 20
- 2.9 Conclusion . . . 22

3 Towards Efficiently Combining Verifiability and Coercion-Resistance **25**
- 3.1 Prerequisites . . . 26
  - 3.1.1 Additional Particular Building Blocks . . . 26
  - 3.1.2 Assumptions on Players and the Adversary . . . 28
  - 3.1.3 Adversarial Uncertainty and a Measure for the Degree of Coercion-Resistance . . . 29
  - 3.1.4 Assessing Coercion-Resistance . . . 30
- 3.2 Protocol by Juels et al. 2005 (JCJ) . . . 32
  - 3.2.1 Description of the Protocol . . . 32
  - 3.2.2 Verifiability . . . 34
  - 3.2.3 Proof-Sketch for Coercion-Resistance . . . 36
- 3.3 SKHS11 Protocol . . . 37
  - 3.3.1 Basic Protocol . . . 38
  - 3.3.2 Enhanced Protocol . . . 41
  - 3.3.3 Proof Sketch for Coercion-Resistance . . . 42
  - 3.3.4 Efficiency and Other Properties . . . 44
- 3.4 SKHS12 Protocol . . . 45
  - 3.4.1 Basic Protocol . . . 46
  - 3.4.2 Enhanced Protocol . . . 48

3.4.3 Proof Sketch for Coercion-Resistance 50
3.4.4 Efficiency and Other Properties 52
3.5 Related Schemes 54
3.5.1 SHKS11 Protocol and CH11 Protocol (Selections) 55
3.5.2 KHF11 Protocol 57
3.6 Conclusion 59

4 More Efficiency Thanks to Hybrid Schemes 61
4.1 Hybrid Schemes 62
4.1.1 Principles 62
4.1.2 Revocation Mechanisms 63
4.2 Protocols for the Internet Channel - SH10 and HS11 66
4.2.1 Protocol Overview 67
4.2.2 Detailed Protocol Definition 68
4.2.3 Security Features 72
4.3 A Proof of Concept for the Electronic Channel of a Hybrid Scheme . . . 73
4.4 Conclusion 74

5 Conclusion 75

Chapter 1

Introduction

To make decisions, many societies like to vote. They agree on voting rules and appoint officials to run the procedures and to draw a bottom-line. If people feel the rules are not respected, they may refuse to accept the bottom-line and watch out for other means to get their way. As a result, a society's stability may be put at stake for no good reason. It is evidently not enough to just enforce the voting rules. People need to be convinced that the rules are actually enforced.

Establishing trust seems rather straight-forward as long as the rules are defined around paper, pens and ballot-boxes. Procedures that entail physical checks and double checks are effective, easy to explain and likely to convince. Any number of officials can be assigned to run the procedures and check that they are running smoothly. Independent observers might even be invited to check on the people who do the checking. If no irregularities are claimed in such a setting, then most likely everything went reasonably well. Too many people would need to engange in fraud actively, which will generally seem too unlikely to believe. If doubts do come up, the procedures can always be re-assessed and adjusted fairly easily.

With Internet voting, lots of the important work happens within the invisible. Computers verify the people's right to vote, count the ballots and account for the secrecy of the ballot to be respected. Above that, it is known that today's standards in Internet technology are not as secure as one might wish for. It is inevitable that engineers make mistakes. These mistakes may cause a lot of damage in case they are found and exploited maliciously. Even rumours about serious intentional security gaps come across the press every now and then.

So if the technology is vulnerable and hard to control, can Internet voting be made trustworthy at all? Moreover, how should the voters be convinced that the voting rules will be respected?

In the past decades, cryptographers have come up with remarkable techniques that allow procedures to be trustworthy even within environments that are possibly insecure. Although the cryptographic building-blocks can not be explained that easily, the approach at bringing them to practice strongly relates to the procedures the people are already familiar with. They *can* be explained.

Towards an analogy, suppose each official is in charge of a special hardware component,

which - unlike the Internet - has been designed for high-security applications. Each device comes from a different vendor. Thanks to cyptography, people can be ensured that everything went fine, unless all officials have colluded or all of their components have been corrupted. Even in the unfortunate case where the security of one of the components will be questioned, the other ones still grant for all votes to be counted correctly and for the secrecy of the vote to remain respected. This very much relates to the security precautions enforced by humans in the traditional settings, where it takes only one loyal and reliable person to expose careless or fraudulent behaviour.

Moreover, by the means of unforgeable cryptographic proofs, independent observers can be allowed to verify that all registered votes have been counted correctly. Even the individual voters can independently verify proofs soundly stating that their vote has been considered in the final tally. Abstainees can verify that their right to vote has not been misused. Remarkably, these proofs are sound, even if all officials collude and all of their components are corrupted. It may seem surprising that verifiability can be offered without compromising the privacy requirements in any way, i.e. without revealing any substantial information that would allow to find out who voted how.

Due to coercion-resistance, even a stronger notion of privacy is added, which may particularly be relevant in countries where any form of remote voting - including voting by mail - will have a hard time finding support. Coercion-resistance allows voters to circumvent attempts that aim at pressurizing them into voting in a given way or into not voting at all. Pressure may be applied by the means of coercion or bribery. Coercion-resistance prohibits coercers and vote-buyers from obtaining a proof on the voters' behaviour, even if the voters choose to deliberately give up their privacy. At first sight, coercion-resistance and verifiability may seem mutually exclusive. How can a proof convince a voter but not a vote-buyer that a vote has been counted in a particular way? - so researchers have wondered.

1.1 Contents of this Thesis

Chapter 2 introduces well-established cryptographic building-blocks that allow Internet voting to be trustworthy. In the course of the chapter, a protocol is introduced and gradually enhanced as new elements are introduced. It ends up in a verifiable protocol that can satisfy very high expectations regarding a system's trustworthiness. In chapter 3, which forms the main part of this thesis, protocols are introduced that additionally satisfy coercion-resistance. They are based on the work in [49], which shows very high ambitions in terms of coercion-resistance. Particularly, [49] proposes a solution that even protects voters from pressure presumably applied by the voting officials. However, the proposal needs to assume computational power on the server side that is impossible to provide when considering political votes with many participants. In effect, it may take weeks, months or years to obtain the final tally. Two of the protocols introduced in chapter 3 have been defined under strong participation of the author of this thesis. The aim lies in reducing the required computation time significantly, without significantly compromising verifiability or coercion-resistance. The prososals are related to other so-

lutions from the literature. It seems that the performance issues can be solved. However, some practical questions related to usability are not addressed in this thesis. Chapter 4 shows ways to achieve coercion-resistance, when assuming that Internet voting needs to be integrated with conventional, paper-based channels. This seems particularly relevant in political voting, where the conventional channels will unlikely be replaced in the near future. Interestingly, it seems that a sufficient notion of coercion-resistance can be introduced much more easily in such a case. Chapter 5 concludes the thesis.

1.2 Contribution

The author of this thesis has worked in collaboration with a team of researchers of the University of Fribourg and the Bern University of Applied Sciences. He has contributed to the following peer-reviewed publications:

- [82], [83] and [74]. (revisited in chapter 3)
- [81], [80], [41] and [26]. (revisited in chapter 4)
- [86] and [84]. (not revisited)

Each of the papers revisited in chapter 3 includes a protocol that solves the performance issue at tallying within verifiable coercion-resistant Internet voting. Still, there seem to be open questions yet to be answered before an application in practice would seem realistic. The results might be combined with other work that addresses usability issues particular to the field.

In chapter 4, [81] presents ways to put coercion-resistance in place by integrating the Internet channel with the traditional voting channels to form a *hybrid scheme*. Further, it presents the requirements on the Internet channel for a hybrid scheme. The papers [80] and [41] each propose a protocol that meets these requirements and that seems a good candidate for an implementation in practice. [26] presents how one of these protocols has been put to practice in a simplified way within the Selectio Helvetica project. The author of this thesis was strongly engaged at defining the system requirements and operating the Selectio Helvetica system during its hosting votes within the Baloti project. In the meantime, Selectio Helvetica has evolved to UniVote and several verifiable student board elections at Swiss universities have been hosted based on the protocol.

In [86], transparency (explaining to which extent trustworthiness is granted), separation of duty (putting multiple officials in charge of differing separated components) and verifiability (issuing proofs that the votes have been processed correctly) are proposed as important measures to ensure trust. [84] looks into the system used in the Norwegian Internet voting pilot in 2011 and analyzes by which means they were implemented. In the meantime, the ordinance of the Swiss Federal Chancellery and its annex take them as preconditions for allowing Internet voting in Switzerland to further evolve [23]. [86] and [84] are not revisited, since they do not focus on the duality of verifiability and coercion-resistance.

Chapter 2

Common Building Blocks to Allow Trustworthy Internet Voting

The following sections summarize the most important building blocks for the results in chapters 3 and 4. These tools are commonly suggested in the literature of Internet voting. The chapter should allow readers with some related background to get acquainted with the challenges and the established cryptographic solutions. An example voting protocol is outlined in section 2.3 and enhanced further along the chapter. This should facilitate the reader's understanding of the building blocks' application within Internet voting and render the subject as a whole more approachable. However, this only works when reading the sections in the proposed order. We start off by giving a few definitions in section 2.1. The introduced notation and conventions are followed throughout the remainder of the document, unless stated otherwise. Section 2.2 introduces the security problems there to solve and section 2.3 shows under which trust assumptions this is meant to be done. Each of the sections 2.4 - 2.8 is dedicated to one building block. This chapter is meant to be self-contained.

2.1 Definitions

We assume an electorate $\mathcal{V} = \{V_1, .., V_N\}$ of N eligible voters and a number N_T of back-end players $\mathcal{T} = \{T_1, .., T_{N_T}\}$, which we call trustees. Each eligible voter V_i is assigned a secret *voting credential* $\mathbf{cred_i}$ which allows to cast a vote. The public counterparts it takes to assess the authenticity of a vote are denoted by $\mathbf{Cred_i}$. If a voter is assigned more than one voting credential, only one is considered *correctly assigned*. The members $\hat{\mathcal{V}} \subseteq \mathcal{V}$ of the electorate express their will by casting one or multiple votes to the trustees in charge (members of $\mathcal{T}$) through a designated channel. The remaining voters abstain. Unlike $\mathcal{V}$, $\hat{\mathcal{V}}$ may be empty, thus assuming the possibility that noone wants to vote. The trustees are in charge of operating the voting system, i.e. distributing the keys, the voting credentials, assembling all cast votes, checking their validity and computing the final tally denoted by Σ. Unless stated otherwise, all system players are considered to have the computational power of an efficient (poly-time) algorithm. As

a consequence, we generally do not model any technical aids, e.g. personal computers, that voters use to perform computations and cast their votes. We also assume the presence of communication channels between the system players, whenever information is passed.

Voters and trustees may be corrupted by an adversary $\mathcal{A}$. His aim might entail breaking the secrecy of the ballot, adding or modifying votes or letting votes disappear. Let $\mathcal{V}^{\mathcal{A}} \subseteq \mathcal{V}$ denote the set of corrupted voters and $\mathcal{T}^{\mathcal{A}} \subseteq \mathcal{T}$ the set of corrupted trustees. We will define further subsets of both $\mathcal{V}$ and $\mathcal{T}$ in the following chapters whenever they are needed.

For a vote to be counted, it needs to be *well-formed* and *legitimate*. These votes are called *valid*, otherwise *invalid*. For a vote to be well-formed, it needs to be taken from a set of N_C predefined voting choices $\mathcal{C} = \{c_1, .., c_{N_C}\}$, where each element of $\mathcal{C}$ refers to one distinct way of filling out a complete ballot in an election or a referendum. Otherwise it is considered *spoiled*. For a vote to be legitimate it needs to be *authentic*, i.e. originate from an eligible voter. Otherwise it is considered *unauthentic*. Additional votes cast by the same voter need to be considered illegitimate, i.e. the vote needs to be *unique*. If a vote is not unique, it is called a *duplicate*. Each system needs to provide a means to check well-formedness, authenticity and uniqueness. We call the process of ruling out unauthentic votes the process of *vote authentication*. All votes that have been cast using a correctly assigned voting credential are considered authentic and pass this step. We thus allow the case where voters hand out their credentials to a third party to vote on their behalf. Clearly, in the lack of a *controlled environment* as identified in [87], it can never be excluded that voters voluntarily hand out their voting credentials to someone else. This problem is inherent to any remote voting scheme, including voting by mail.

Let v^{x} denote a generic set of all cast votes that hold an attribute x. The set can be empty, otherwise $\mathrm{v}^{\mathrm{x}} = \{v_1^x, .., v_{n_x}^x\}$. Its attribute can be specified by choosing x as $well-formed$, $legitimate$, $authentic$, $unique$ or $valid$ thus giving it the corresponding meaning. The set $\mathrm{v} = \{v_1, .., v_n\}$ denotes the entire collection of cast votes (we thus allow to omit $x = cast$).

2.2 Security Requirements

Internet voting needs to be assessed in the light of security requirements and trust assumptions that vary across societies and time. Even when focusing on a specific ballot (e.g. the parliamentary elections in a given country), the appropriate technical requirements can hardly be derived from existing legislation. Also, it is difficult to anticipate the nature of the public's doubts (to be compensated for by making more restrictive requirements) and the willingness to trust (allowing to waive further requirements for the benefit of efficiency or user-friendliness). Maybe due to these uncertainties, a lot of scientific work proposes cryptographic protocols without any refinement to specific contexts of application. Instead, the ambition seems to lie in satisfying very strong requirements while allowing only very weak trust assumptions.

The following requirements are strongly based on [39], whereas we have made some changes which are captured within the explanations. We emphasize that the terms are defined in a way to reflect how they are often used in the technical literature. Some definitions differ from the common use in legislation or other areas of research. Also their use in the technical literature is far from uniform, we refer to [48] for a good overview.

Democracy: A system is democratic if only eligible voters can vote (*eligibility*) and if only one vote is counted per eligible voter (*uniqueness*). This requirement formalises a basic princple generally adhered to in political voting. However, it would need to be generalized when it comes to certain non-political contexts, e.g. shareholder votes in private companies where votes are weighed in a function of the number of shares kept by the voters. We take this requirement from [39] without changing it.

Accuracy: A system is accurate if cast votes are not altered (*integrity*), valid votes are considered in the final tally (*completeness*) and invalid votes are not counted (*soundness*). We deviate from the description in [39], where integrity, completeness and soundness are introduced as requirements that a system must render impossible not to comply with (e.g. *A system is accurate if cast votes* ***can not be*** *altered [../]*). We weaken the requirement, since many protocols do not consistently explicit any roll-back mechanisms, based on observation by the voters and mutual observation among the trustees (although generally this would be possible). However, we point out that the requirement *verifiability* defined further down ensures that accuracy can not be breached **unnoticed**.

Secrecy: A system respects secrecy if no cast vote can be linked to its voters (*vote-privacy*) and if no intermediate results can be obtained before the voting period ends (*fairness*). We describe vote-privacy the same way as *anonymity* is described in [39]. It seems that *anonymity* is the perfect term to be employed for the corresponding requirement outlined next. It has been used the same way in [41]. The term *vote-privacy* has commonly been used in the sense of our definition, as for instance in [25]. We use the term *secrecy* to hold both *vote-privacy* and *fairness*. This has previously been proposed in [86], whereas *vote-privacy* there is called *privacy*. We thus keep the term *privacy* to accomodate *vote-privacy*, *anonymity* and *coercion-resistance*, which all relate to the privacy of the individual voters.

Anonymity: A system respects anonymity if none of the system players are able to elicit which voters have cast a vote. The term has been prosposed in this sense in [44]. It is not identified in [39]. On one hand it can be beneficial for vote-privacy (consider a vote with few participants) and it is a strict pre-condition for coercion-resistance. Additionally we must wonder if voters will agree to have an IT-system contain information on whether they have participated or abstained at a vote. One may argue that in traditional voting schemes, the staff at the polling station can witness

which voters cast their votes, thus anonymity should not be considered a security issue. However, especially along with verifiable voting schemes that inherently need to allow access to a lot of data, it might seem imperative to explicitly require anonymity, in order to at least ensure anonymity towards the broad public. In any case, the problem needs to be addressed. The protocols outlined in chapters 3 and 4 are designed to meet this requirement under very weak trust assumptions. Finally, we point out that some legislations oblige the voters to participate at votes. In order for the authorities to indentify abstainees, anonymity must clearly not be granted to the greatest possible extent.

Verifiability: A system is verifiable if the system players are able to verify the accuracy of a voting process independently by the combined means of *individual verifiability* and *universal verifiability*.

- **A system is *individually verifiable* if voters can verify that their own votes have been considered correctly in the final tally.** This requirement has been taken unchanged from [39] and reflects the definitions often found in the literature, as in [48], [59], [58]. The literature sometimes reduces the scope of individual verifiability and summarizes it under two distinct verifiability conditions. First, voters should be able to verify that their vote has been *cast-as-intended* and second to verify that it has been *recorded-as-cast*.[1] These terms are widely proposed, for instance in [34], [86], [2] and [50]. By these notions, the scope of individual verifiability is not fully covered according to our definition, since voters are still not able to verify whether their recorded vote is further correctly processed and considered in the final tally. This can be compensated for by additionally defining *counted-as-recorded* verifiability as a third condition for individual verifiability. This term (or a variant such as *tallied-as-stored*) is often proposed, for instance in [34] and [58]. However, some literature also proposes to define individual verifiability as a synonym for merely *cast-as-intended* and *recorded-as-cast*, as in [1] or [34]. Systems that offer individual verifiability in the sense of *cast-as-intended*, *recorded-as-cast* and *tallied-as-recorded* are also often called *end-to-end* verifiable as in [70] and [58]. However, by requiring universal verifiability as defined further down, *counted-as-recorded* verifiability and thus the full scope of individual

[1] The distinction allows to group different types of attackers, which in practice follow different strategies that require different counter-strategies: A voter who verifies that his vote has been cast as intended knows that his vote has been sent out unchanged. As an example, a corrupted trustee T_1 in charge of supplying him with the voting software has obviously not succeeded in modifying the vote. A voter who verifies that his vote has been recorded as cast knows that T_2 in charge of recording the vote has not succeeded in modifying (or not recording) it either. However, since in our model we assume that voters have poly-time computational capacity, the voters V_i will always be able to memorize all steps it took from choosing their vote intention v_i^{valid} up until the cast-as-recorded verification. In the context of this thesis, the distinction between these terms is therefore hardly ever relevant. (If distinctions between voters and their computers are yet made, the computers are either considered trustworthy or they need to be modeled as a trustee potentially corrupted by the adversary. In the latter case the distiction between the two verification steps can make sense. We also refer to the so-called trusted-platform-problem which is brought up in the next section.)

verifiability is also implied in the sense of our definition. The same argument is followed in [34] and [50]. When considering individual verifiability, it therefore seems legitimate only to consider the voters' ability to verify that their votes were *cast-as-intended* and *recorded-as-cast*, as long as universal verifiability is granted.

- **A system is *universally verifiable* if the voters can independently verify that all recorded votes have been counted correctly in the final tally.** Taking advantage of the individual verifiability property of a system requires prior knowledge that only the *individual voters* have (their voting intention and the information meant to be recorded). However, universal verifiability allows all voters to witness the correct processing of each recorded vote, regardless of any prior knowledge, hence *universally*. We have rephrased this requirement taken from [39] in order to avoid an ambiguity.[2] The term is widely used in this sense, as in [59] (*strong universal verifiability*), [12], [58] and [50]. Based on our definition of the term *valid vote* (as a vote which is well-formed, authentic and unique), we may also specify universal verifiability as follows: *universal verifiability allows voters to verify that the final tally reflects the consolidation of all recorded valid votes and none of the invalid votes*. Some literature further identifies *eligibility verifiability* as a verifiability requirement that is covered by universal verifiability in our definition. This is for instance done in [79]. In these cases *universal verifiability* is weakened, thus considering protocols that do not provide eligibility verifiability. Thus, universal verifiability becomes comparable with *average universal verifiability* in [58]. Since the term is quite new and because some protocols that do not satisfy eligibility verifiability may yet satisfy more than just a weakened sense of universal verifiability, we do not make this distiction in this thesis. Instead, we will show in which way a protocol does not satisfy universal verifiability whenever it may be the case.

Coercion-resistance: A system is *coercion-resistant* if voters are not able to waive their vote-privacy and their anonymity and prove having done so. The term has not been identified in [39], since it assumes a setting where coercion is not a problem. Coercion-resistance is a stronger version of both vote-privacy and anonymity. It has been formally defined in [49] and aims at rendering both coercion and bribery infeasible. As pointed out in [16] there is a difference to the two concepts, since coercers punish and vote-buyers reward. Yet, both involve having voters reveal their voting behaviour. We refer to [57] for a good overview of further known definitions. In the past, the requirement *receipt-freeness* alone was proposed as the means to circumvent coercion and vote-buying. A receipt is considered the information it takes for a coercer

[2] According to [39] voters should be able to verify the correct consideration of every *cast* vote. This is due to the irrelevance of distinguishing between cast and recorded votes in the context of that paper. In order to be in line with the terminology introduced above in the context of individual verifiability, we need to relate universal verifiabity to the set of *recorded* votes. Also note, that this definition implies *counted-as-recorded* verifiability, by the means of which each voter can verify that *his* recorded vote is correctly tallied.

to figure out how voters voted, as in [15], [10], [81] and [80]. The absence of receipts ensures vote-privacy, even for voters who wish to comply with an adversary.[3] However, as argued in [49], receipt-freeness is not a satisfying condidion for rendering coercion attacks infeasible as long as coercers are able to verify whether voters participated at a vote or not. Therefore, coercion-resistance as defined in [49] also aims at rendering so-called *forced-abstention* attacks infeasible and requires special protection of the voters' anonymity. In the sense of this definition, certain protocols may not be considered coercion-resistant, although they may have been when following other definitions. For instance [66] is attributed to be coercion-resistant in [48], although voters can always sell their voting credential to the coercer, thus giving him full control.

2.3 Trust Assumptions

Not all security requirements can be met simultaneously without assuming any trust. Since they form the base-line for assessing a system's trustworthiness, it is crucial to clearly identify compromises. These compromises can be captured in terms of trust assumptions in the involved players. The trustworthiness of an applied system then depends on the trustworthiness of these players. Their trustworthiness in return depends on their concrete implementation. Clearly, if a trustee can be implemented in a way that he is perceived to be uncorruptable, very simple and efficient voting protocols would suffice. Example 2.3.1 outlines such a protocol.

Example 2.3.1 *Assume that each voter has initially securely exchanged public signature keys with trustee T_1. Voters use their private signature keys as their voting credential. T_1 has initially constructed the private key for decrypting votes and forwarded the corresponding public key to the voters. To cast a vote, voters encrypt and sign their vote and send it to T_1 through an insecure channel. Upon reception, T_1 checks the signature and whether the voter has previously cast a vote. Upon success, he decrypts and adds the vote to the tally. Since their vote was sent through an insecure channel, the voters ask T_1 for a confirmation by sending him a challenge. To confirm that a vote has been counted, T_1 sends the voter a signature of the challenge. After the voting period ends, T_1 outputs the final tally.*

This protocol complies with democracy, accuracy and secrecy, but only if we assume T_1 to be trustworthy. Can as setting be found that would justify T_1's trustworthiness? Assume that T_1 were a machine and a person in charge of it. Questions arise. Is this a trustworty person? Is he competent? Are other people granted access to the machine? How is it protected? Can people access the machine unnoticed? What if the software is malfunctioning? How was the machine set up and by whom?

[3] We exclude the case where an adversary breaks vote-privacy just on the basis of the final tally Σ (for example when all voters vote and all vote the same) or prior knowledge about its distribution (thus allowing better guesses about the voting behaviour of individual voters). We therefore assume adversarial uncertainty regarding Σ as introduced in the following section.

Some of these questions would also be asked in traditional voting, when assuming that one single person receives all ballots and performs the full tallying and counting procedures on his own. An important part of the solution there is to hire a suffiently large number of well-reputed people who inherently observe eachother's actions. Similarly in our example protocol, a group of individuals could be put in charge of operating and surveiling the machine - as if the machine were a physical ballot-box. Although the machine is easy to observe physically, it is not so easy to dectect malicious accesses from the Internet or even from inside the infrastructure. Also, it is hard to judge whether the software runs correctly and whether there are intentional or unintentional backdoors for attacks. Therefore Internet voting protocols generally propose multiple separated peers to perform a given action. Indeed, the prominent methods in Internet voting propose to apoint a number of multiple trustees, which in practice are meant to be realized by distinct people who use distinct technical aids.

Now the overall system's trustwortiness depends not on just one, but a whole set of trustees. Although it may seem that more players will generally cause more trouble, appropriate protocols can ensure resistance against malicious players. Particularly, if a sufficient number of trustees perform their actions according to the protocol, the remaining trustees *cannot* negatively affect the operations in any way. Even if a trustee looses or hands out all his data or even if he intentionally produces false outputs, it is sufficient to have a subset of trustees that perform well. The trustworthiness of the overall system can then be expressed in terms of the minimal number of trustees that need to be trusted.

The following paragraphs exhibit the trust assumptions regarding each system player and relate them to the security requirements. They reflect to which extent proposals from the literature manage to meet the security requirements defined in the previous section. This serves as a benchmark for assessing the protocols disussed in chapters 3 and 4. As mentioned above, we do not seek for maximally reduced trust assumptions when it comes to guaranteeing accuracy. Instead, we focus on verifiability, which is there to provide evidence that accuracy was indeed respected. This leaves us with adversaries who try to break accuracy *unnoticed*, privacy and fairness. We now present the weakest trust assumptions regarding trustees and voters found in the literature.

Trustees: Regarding adversaries who want to break accuracy unnoticed, verifiability should be in place, such that no trust in *any* trustee is required. Regarding adversaries who want to break the secrecy requirements or anonymity, at least a majority of trustees needs to be trustworthy in order for the attack to fail.[4] Sometimes the trustees are split into groups, such that a majority of each group is assumed trustworthy. Although these conditions allow to achieve secrecy and anonymity, coercion-resistance requires stronger assumptions. For coercion-resistance, all *registrars* (a specified subgroup of trustees) inherently need to be trustworthy at the stage where a voter obtains his voting

[4]Theoretically one could go as far as assuming only *one single* trustworthy player among all trustees, as the weakest permissible assumption. However, this is usually not suggested for the sake of robustness. This issue is further disussed in section 2.7.

credential. During the remaining phases of the protocol, the voter needs to know which trustee he can continue to trust in each subgroup.

Voters: Verifiability should be granted to each voter even when all of the remaining voters are corrupted. However, as with the trust assumptions regarding the trustees, privacy can not be demanded uncoditionally, i.e. without assuming *any* uncorrupted players. Imagine that all voters are corrupted, except for one. In this case the adversary can easily break that voter's vote-privacy, by using the knowledge from the voters he controls and the final tally Σ. Similarly, if all voters vote the same, Σ reveals to the adversary that each voter either abstained or voted for the winner of the election. As also for coercion-resistance, we always need to be able to assume a sufficient degree of prior *adversarial uncertainty* regarding Σ, which in return implies the need to assume a sufficiently large set of uncorrupted voters. This reflection and the term *adversarial uncertainty* can also be found in [49]. In the case of coercion-resistance this condition is even strengthened for the following reason. Voters need to be able to simulate casting a vote that complies with the coercer's demands, while misleading the coercer with regard to its authenticity. In the extreme case, voters should be able to hand out a fake voting credential that cannot be told apart from a correctly assigned one. Clearly, the set of unauthentic votes cast with such a credential needs to be ruled out in the course of vote authentication. In verifiable systems the adversary will learn the number Γ of such votes. Therefore, we require adversarial uncertainty not just regarding Σ, but also regarding Γ. In order to achieve adversarial uncertainty, the number of corrupted voters needs to be limited.

In the following sections we introduce the building blocks commonly known in the literature as we gradually adapt the example protocol from above. In its final state it will satisfy democracy, accuracy, vote-privacy, fairness and verifiability. Anonymity and coercion-resistance are left for the chapters that follow.

2.4 Public Bulletin Boards

Public bulletin boards ($\mathcal{PB}$) are public broadcast channels run by a group of trustees that grant for their integrity, robustness and authenticity. In the Internet voting literature a $\mathcal{PB}$ serves as an instrument for verifiability, since it stores all votes and it is publicly accessible. Technically, the voters cast their vote by appending it to $\mathcal{PB}$. A designated protocol ensures that no entries can be removed unless a majority of trustees give their consent. Voters can verify that their vote is cast as intended and recorded as cast. In many cases the electoral register and the public counterparts $\mathbf{Cred_i}$ of the voting credentials $\mathbf{cred_i}$ are also published on $\mathcal{PB}$, which allows everyone to witness that only legitimate votes are recorded. $\mathcal{PB}$ can also be used to publish the proofs that allow for universal verifiability. [69] describes a way to implement a secure $\mathcal{PB}$.

By enhancing the example protocol from the previous section by the use of a $\mathcal{PB}$, we enable voters to verify that their vote has been cast as intended and recorded as cast,

without assuming T_1 to be trustworthy. Sending T_1 a challenge for him to sign after casting a vote can thus be waived.

2.5 Homomorphic Randomized Encryption and ElGamal Cryptosystem

Complying with the secrecy requirements envolves encrypting the votes prior to sending them to the voting servers. A crucial aspect of any voting protocol lies in the choice of the appropriate cryptosystem. In order to comply with fairness, it must ensure that encryptions reveal no information on the plaintext votes, even when there are only few voting options. In order to ensure vote-privacy throughout the voting procedures, decrypted votes may not be linkable to the voters they were cast by. Yet, for the sake of accuracy any system needs to provide a means for deciding whether a vote is to be counted or not, based on a pre-established electoral register.

In order to *detach votes from voters* many protocols rely on the ability to perform computations in the ciphertext space, i.e. without decrypting the votes. Taking advantage of the homomorphic property of the well-known cryptosystems ElGamal [28] and Pailler [67] offers one way of performing the appropriate computations. The protocols introduced in chapters 3 and 4 rely on this approach. As often done in the literature, we use ElGamal to exemplify the concept of applying a homomorphic randomized cryptosystem in this context. Still we point out that also non-homomorphic encryption algorithms have been proposed for Internet voting, e.g. salted RSA can be used for schemes that are based on blind signatures, as in [31]. However, in [31] verifiability is reduced, since a collusion of trustees would be able to cast and count unauthentic votes unnoticed.

Let $E(m_1)$ and $E(m_2)$ be encryptions of plaintexts m_1 and m_2 respectively, then due to the homomorphic property $E(m_1) \circ_A E(m_2) = E(m_1 \circ_B m_2)$, where $\circ_A$ and $\circ_B$ are two operations, which are instantiated further down for ElGamal.

The ElGamal cryptosystem is based on a multiplicative cyclic group $(G, \cdot)$ of finite order q, for which the decisional Diffie-Hellman assumption (DDH) is believed to hold. A safe choice for $(G, \cdot)$ is a subgroup $\mathbb{G}_q \subset \mathbb{Z}_p^*$ of order $q = (p-1)/k$, where p and q are large primes.[5] An implication of the DDH-property of $(G, \cdot)$ is informally speaking that ElGamal-ciphertexts reveal no information on the encrypted plaintext, even when a known domain of only two possible plaintexts is given ($\mathrm{IND} - \mathrm{CPA}$).[6] This security

[5] Under the DDH assumption it is impossible to construct an efficient algorithm with a non-negligible advantage (over a random coin's decision) at deciding whether dealing with a *Diffie-Hellmann tuple* $(g^a, g^b, g^{a \cdot b})$ or a triplet of random numbers (g^a, g^b, g^c), where $g \in \mathbb{G}_q$ is given and $a, b, c \in_R \mathbb{Z}_q$. The advantage over letting a coin decide whether a triplet is a Diffie-Hellmann tuple drastically falls (to negligibility) in the bit-length of p and q. Clearly, DDH also implies the discrete logarithm assumption (DL), under which it is impossible to construct an efficient algorithm with a non-negligible advantage at computing a, when given g and g^a, where $g \in \mathbb{G}_q$ and $a \in_R \mathbb{Z}_q$.

[6] As per the definition of the $\mathrm{IND} - \mathrm{CPA}$ property, ElGamal provides **ind**istinguishability under **c**hosen **p**laintext **a**ttack. The $\mathrm{IND} - \mathrm{CPA}$ property stands for the assumption, that no efficient algorithm (an

feature of cryptosystems has already been identified in 1984 [36]. Nowadays modern cryptographic protocols commonly rely on it.

The public parameters of an ElGamal cryptosystem are p, q, and a generator g of $\mathbb{G}_q$. An ElGamal key pair is a tuple (d, e), where $d \in_R \mathbb{Z}_q$ is the randomly chosen private key and $e = g^d \in \mathbb{G}_q$ is the corresponding public key. If $m \in \mathbb{G}_q$ denotes the message to encrypt, then the pair $(x, y) = (g^k, m \cdot e^k)$ is the encryption of m with randomness $k \in_R \mathbb{Z}_q$. Note, that the IND – CPA property of ElGamal relies on choosing a fresh randomness at every encryption. For a given ElGamal encryption (x, y), m can be recovered by computing m as $\frac{y}{x^d}$.

Instead of using $\mathbb{G}_q$ as the domain of m and $\mathbb{G}_q \times \mathbb{G}_q$ as the image space, ElGamal can also be defined over elliptic curves, such that m, x and y are points on the curve. This can bring an increase in efficiency. However, we will adhere to the more classical way of describing ElGamal as it is often found in the Internet voting literature. ElGamal is homomorphic since both aforementioned operations $\circ_A$ and $\circ_B$ can be instantiated by *multiplication* (component-wise when multiplying ciphertexts) modulo the value p to satisfy the condition for homomorphisms. If not stated otherwise, we will use $E(m, k)$ to denote an ElGamal encryption of m under randomness k. The text will suggest which public key is being used. Sometimes we will use the shorter notation $E(m)$ if the randomness's value is irrelevant.

We are now ready to enhace the example protocol in 2.5.1. It shows how the homomorphic property can be used at tallying the votes. Such *homomorphic tallying* is further explained in [21] and adopted in [45]. This version of the protocol forms another step towards verifiability.

Example 2.5.1 *We consider a yes-no referendum, where* $\mathcal{C} = \{1, h\}$ *and* $h \in \mathbb{G}_q \setminus \{1\}$ *denotes a vote for* yes, 1 *denotes a vote for* no. *Before casting votes, we assume that the voters' identities* $V_1, .., V_N$ *are published on* $\mathcal{PB}$ *along with their public signature keys* $\mathbf{Cred_i}$. *Let* $E_e(v_1, k_1), ..., E_e(v_n, k_n)$ *be the collection of all ElGamal encrypted votes cast to* $\mathcal{PB}$ *along with their signatures. For now, we assume that they are all well-formed. Since it is publicly known which voters cast which ciphertext,* T_1 *may not apply private key* d *to decrypt and publish each vote separately - doing so would breach vote-privacy in front of the broad public. Instead,* T_1 *multiplies all legitimate ciphertexts to obtain* $E(v) = E(\prod_{i=1}^{n_{valid}} v_i^{valid}, \sum_{i=1}^{n_{valid}} k_i^{valid})$, *i.e. due to the homomorphic property the encryption of all multiplied plaintexts. After decryption using private key* d, *it is easy to try out values for* n_{yes} *to satisfy the equation* $h^{n_{yes}} = v \pmod p$. *The value* n_{yes} *is the number of yes-votes. The number of no-votes is* $n_{valid} - n_{yes}$.

---

adversary) with a non-negligible advantage at winning the IND – CPA *game* can be constructed. The IND – CPA game is defined as follows: First, the adversary receives the public parameters from the challenger. He may perform an arbitrary number of computations. Then, he forwards two plaintexts to the challenger. The challenger randomly picks one of both plaintexts and sends an encryption of it back to the adversary. The adversary may perform further computations, before he eventually guesses which plaintext the encryption contains. As shown in [85] DDH and IND – CPA of ElGamal imply eachother.

The voters can now verify the correctness of T_1's assessment of the votes' validity and his summing up the votes. Two problems remain to be solved before the protocol satisfies the verifiability requirement. First of all, voters can not yet verify that T_1 decrypts the votes correctly, i.e. using his private ElGamal decryption key d. T_1 can convince the voters by publishing a *zero knowledge proof* as introduced in the next section. The second problem concerns voters who cast double or multiple votes, e.g. by choosing $v = h^2$ or h^{-2}. By publishing a zero-knowledge proof on $\mathcal{PB}$, they can unambiguously prove that the posted ciphertext encrypts a well-formed vote, i.e. a vote from the set of possible choices $\mathcal{C}$.

Regarding secrecy, no improvements have yet been made since the first version of this protocol, as T_1 can still use d to decrypt the individual votes to break vote-privacy or to break fairness when doing so prematurely. This problem will be addressed in section 2.7. However, in the enhanced example, no compromises regarding the secrecy requirements had to be made either, although we are just a step away from complying with verifiability.

2.6 Non-Interactive Zero-Knowledge Proofs

A zero-knowledge (ZKP) proof allows a prover $\mathcal{P}$ to demonstrate to a verifier $\mathcal{V}$ that a statement is true, without revealing anything but the truth of the statement itself. A particular class of ZKP are so-called *proofs of knowledge*. In a simple case, the prover might just want to demonstrate that he knows the preimage ω of a public value $x = \phi(\omega)$, where $\phi : \mathbb{G} \rightarrow \mathbb{H}$ is a candidate one-way function. Due to *zero-knowledge*, the prover has a tool for proving his knowledge without allowing the verifier to learn anything about ω at all. Specifically, a zero-knowlege proof must satisfy *completeness* (in the case of an honest prover, an honest verifier accepts the proof), *soundness* (in the case of a corrupted prover, an honest verifier does not accept the proof) and *zero-knowledge* (no verifier learns anything but the truth of the statement itself).

So-called Σ-protocols are special proofs of knowledge where the preimage space $\mathbb{G}$ is finite and the function ϕ is a homomorphism. Algorithm 1 shows a Σ-protocol due to [75], where we define $\mathbb{G} := \mathbb{Z}_q$, $\mathbb{H} := \mathbb{G}_q$ and $x = \phi(\omega) := g^\omega$.

It is easy to see that the protocol is *complete*. For an understanding to which extent it complies with *soundness*, consider the following experiment, where we assume that $\mathcal{P}$ has an initial strategy to convince the verifier without knowing ω. He applies his strategy and chooses a value for t of which he may or may not know the preimage $r = \log_g t$. Then he receives the challenge c, computes s and sends the value to $\mathcal{V}$. Now $\mathcal{V}$ rewinds $\mathcal{P}$ through black-box access and sends him another challenge c'. $\mathcal{P}$ computes the corresponding value s' and sends it to $\mathcal{V}$, who now has the two transcripts (t, c, s) and (t, c', s'). Now $\mathcal{V}$ learns ω by computing $\frac{s-s'}{c-c'}$. Since $\mathcal{V}$ found out ω by communicating with $\mathcal{P}$, obviously $\mathcal{P}$ had to know it in the first place, which contradicts the initial assumption. Apparently, the prover cannot convince the verifier without knowing (by guessing) the challenge c beforehand. Indeed, if the prover *does* know c, he is able to simulate an accepting transcript (t, c, s) of the protocol by choosing $s \in_R \mathbb{Z}_q$ and

Algorithm 1
ZKP ($\mathcal{P} \leftrightarrow \mathcal{V}$): $ZKP[(\omega) : x = g^{\omega}]$
(Prove and verify knowledge of discrete logarithm ω, where $\omega = \log_g x$)

Require: $\mathcal{P}$ knows p, q, g, x, ω; $\mathcal{V}$ knows p, q, g, x
$\mathcal{P}$: $r \leftarrow \text{random}(\mathbb{Z}_q)$
$\mathcal{P}$: $t \leftarrow g^r$
$\mathcal{P}$: send t to $\mathcal{V}$
$\mathcal{V}$: $c \leftarrow \text{random}(\mathbb{Z}_q)$
$\mathcal{V}$: send c to $\mathcal{P}$
$\mathcal{P}$: $s \leftarrow r + c \cdot \omega$
$\mathcal{P}$: send s to $\mathcal{V}$
$\mathcal{V}$: if $\mathbf{t} \cdot \mathbf{x^c} = \mathbf{g^s}$ then **accept**; else reject

computing t as $\frac{g^s}{x^c}$. The prover's probability of guessing c correctly (the *knowledge-error*) is $\frac{1}{q}$, which is negligible in the bit-length of q. To show that the protocol is *zero-knowledge*, we assume that the verifier can only be passively corrupted, i.e. he may try to learn ω based on the transcripts but without deviating from the protocol, particularly by not choosing c at random. This notion is captured as *honest verifier zero-knowledge* (HVZK) in [8] or *special honest verifier zero-knowledge* in [56]. To argue that such a verifier learns nothing about ω when following the above protocol, we note that he is able to simulate any accepting transcript, just as the prover can when knowing c. Since any such transcript is potentially a real transcript with equal probability, the verifier can not learn anything additional by performing the protocol with the prover.

Interactive zero-knowledge proofs as in the example above are *not transferable* from one verifier to another, since no third party can distinguish whether the prover simply simulated a transcript or not. To this end, we replace the verifier's role of choosing the challenge c, by having the prover compute it himself as $c \leftarrow \mathcal{H}(t)$, where $\mathcal{H}(\cdot)$ is implemented as a collision-resistant hash-function. The proof thus becomes non-interactive and any third party can play the role of the verifier at accepting or rejecting a transcript. This strategy, i.e. allowing the implementation of the verifier's random number generator as a hash-function, follows the Fiat-Shamir heuristic proposed in [30]. Such *non-interactive zero-knowledge proofs* (NIZKP) are sound and HVZK in the random oracle model (ROM), which was proposed in [9]. ROM allows to model $\mathcal{H}(\cdot)$ as a random oracle, which returns a truly random value from a given image-space upon reception of an input value. Like for hash-functions, the returned random value is always the same for a given input. Then, by making a few further assumptions on how $\mathcal{P}$ and the random oracle can be accessed, efficient simulators can be constructed that are able to extract the witness ω (to justify soundness) or to simulate a valid transcript (to justify witness-hiding, i.e. HVZK). Since our security arguments in the following chapters rely on the random oracle model, we will assume the existence of such simulators and thus the ideality of ZKP.

As shown in [56] and [11], more general Σ-protocols can be defined to prove knowledge

of complex relations. Their non-interactive versions are also secure in the random oracle model in the above sense. One example is proving knowledge of a value ω, and proving that it satisfies the condition of being the discrete logarithm of two different function values $x_1(=g_1^{\omega})$ and $x_2(=g_2^{\omega})$ at the same time. This is often captured as the notion of proving the *equality of discrete logarithms*, as proposed in [14]. In this case, the prover does not only demonstrate his knowledge but also the fact that such a value actually exists at all. Σ-protocols also allow a prover to demonstrate his knowledge of at least one out of two preimages ω_1 and ω_2 of two different function values $x_1 = g^{\omega_1}$ and $x_2 = g^{\omega_2}$, without revealing which one he actually knows ($\mathcal{OR}$-proof). In order to express what is being proved, we use the intuitive *ZKP-notation*. The first example can be expressed by $ZKP[(\omega) : (x_1 = g_1^{\omega}) \wedge (x_2 = g_2^{\omega})]$ and the second one by $ZKP[(\omega) : (x_1 = g^{\omega}) \vee (x_2 = g^{\omega})]$. Basically the knowledge of any disjunction or conjunction of linear relationships between preimages can be proved with Σ-protocols. In algorithms 2 and 3 we settle for showing the examples of equality of discrete logarithm and the $\mathcal{OR}$-proof, both mentioned above. They are commonly employed in various voting protocols and form the basis of the two missing elements for the next enhancement of our protocol shown in example 2.6.1.

Algorithm 2
NIZKP $(\mathcal{P} \rightarrow \mathcal{V})$: $ZKP[(\omega) : (x_1 = g_1^{\omega}) \wedge (x_2 = g_2^{\omega})]$

Require: $\mathcal{P}$ knows $p, q, g_1, g_2, x_1, x_2, \omega$; $\mathcal{V}$ knows p, q, g_1, g_2, x_1, x_2
$\mathcal{P}$: $r \leftarrow \text{random}(\mathbb{Z}_q)$
$\mathcal{P}$: $t_1 \leftarrow g_1^r$; $t_2 \leftarrow g_2^r$
$\mathcal{P}$: $c \leftarrow \mathcal{H}(t_1|t_2)$
$\mathcal{P}$: $s \leftarrow r + c \cdot \omega$
$\mathcal{P}$: send t_1, t_2, c, s to $\mathcal{V}$
$\mathcal{V}$: if $\mathbf{t_1} \cdot \mathbf{x_1^c} = \mathbf{g_1^s}$ and $\mathbf{t_2} \cdot \mathbf{x_2^c} = \mathbf{g_2^s}$ and $\mathbf{c} = \mathcal{H}(\mathbf{t_1}|\mathbf{t_2})$ then **accept**; else reject

Algorithm 3
NIZKP $(\mathcal{P} \rightarrow \mathcal{V})$: $ZKP[(\omega) : (x_1 = g_1^{\omega}) \vee (x_2 = g_2^{\omega})]$

Require: $\mathcal{P}$ knows $p, q, g_1, g_2, x_1, x_2, \omega_2$ (where $x_2 = g_2^{\omega_2}$); $\mathcal{V}$ knows p, q, g_1, g_2, x_1, x_2
$\mathcal{P}$: $s_1 \leftarrow \text{random}(\mathbb{Z}_q)$; $c_1 \leftarrow \text{random}(\mathbb{Z}_q)$
$\mathcal{P}$: $t_1 \leftarrow \frac{g_1^{s_1}}{x_1^{c_1}}$ // (t_1, c_1, s_1) is a simulated proof for knowing $\log_{g_1} x_1$
$\mathcal{P}$: $r_2 \leftarrow \text{random}(\mathbb{Z}_q)$
$\mathcal{P}$: $t_2 \leftarrow g_2^{r_2}$
$\mathcal{P}$: $c \leftarrow \mathcal{H}(t_1|t_2)$; $c_2 \leftarrow c - c_1$
$\mathcal{P}$: $s_2 \leftarrow r_2 + c_2 \cdot \omega_2$
$\mathcal{P}$: send (t_1, c_1, s_1), (t_2, c_2, s_2) to $\mathcal{V}$
$\mathcal{V}$: if $\mathbf{t_1} \cdot \mathbf{x_1^{c_1}} = \mathbf{g_1^{s_1}}$ and $\mathbf{t_2} \cdot \mathbf{x_2^{c_2}} = \mathbf{g_2^{s_2}}$ and $\mathbf{c} = \mathcal{H}(\mathbf{t_1}|\mathbf{t_2})$ and $\mathbf{c} = \mathbf{c_1} + \mathbf{c_2}$ then **accept**; else reject

We are now ready to apply these two algorithms in the example. In this version the protocol is verifiable and essentially describes the one presented in [20].

Example 2.6.1 *All voters prove that their vote* $E(v_i) = (x_i, y_i) = (g^{k_i}, v_i \cdot e^{k_i})$ *is well-formed, i.e.* $v_i \in \{1, h\}$. *They do so by posting a non-interactive zero-knowledge proof*

$$ZKP[(\omega) : (x_i = g^\omega \wedge y_i = e^\omega) \vee (x_i = g^\omega \wedge \frac{y_i}{h} = e^\omega)] \tag{2.1}$$

to $\mathcal{PB}$ *along with it.* T_1 *then only considers legitimate votes as valid if the corresponding proof holds. In order for* T_1 *to decrypt the product of valid votes* $E(v) = E(\prod_{i=1}^{n_{valid}} v_i^{valid}, \sum_{i=1}^{n_{valid}} k_i^{valid}) = (x, y) = (g^k, v \cdot e^k)$, *he computes the value* $X = x^d$ *and posts it to PB along with the following proof:*

$$ZKP[(\omega) : (X = x^\omega \wedge e = g^\omega)] \tag{2.2}$$

Finally, T_1 *computes* v *as* $\frac{y}{X}$ *and publishes the outcome of the vote.*

The protocol is now universally verifiable, since voters can assess the well-formedness (by verifying the first zero-knowledge proof) and the legitimacy of all encrypted votes cast to $\mathcal{PB}$. They can re-compute the ciphertext $E(v)$ themselves and witness that all encryptions of valid votes are taken into account and no invalid ones. By verifiying the second zero-knowledge proof they witness that the value X it takes for decrypting $E(v)$ is correctly obtained. From that point on, anyone can compute the outcome of the vote and compare it with the result published on $\mathcal{PB}$. The protocol is also individually verifiable, since voters can witness that their vote is displayed on $\mathcal{PB}$.

Despite verifiablity, noone can obtain any information from $\mathcal{PB}$ that would help at decrypting individual votes. However, T_1 can still do so. The next section shows how such a single point of failure regarding the secrecy requirements can be avoided, i.e. in the strict sense of the previously outlined trust assumptions.

2.7 Secure Multiparty Computation

Secure multiparty computation (MPC) provides a tool that allows to distribute secrets among a set of multiple trustees $\mathcal{T} = \{T_1, .., T_{N_T}\}$. These techniques can be applied in Internet voting to ensure that no single entity is in possession of the private key d it takes to violate vote-privacy or fairness. Zero-knowledge proofs are commonly employed within MPC-protocols to detect faulty computations and thus ensure verifiability.

Secret sharing dates back to Shamir's secret sharing scheme in 1979 [77]. Already this first step towards secure MPC-protocols took into consideration that a large number of trustees is more likely to compromise robustness if correctly reconstructing the secret were to depend on the correct behaviour of each single trustee. The proposal allows to circumvent this problem by allowing a predefined number t of less than N_T trustees to obtain the secret. Such protocols are called MPC with a threshold t or (t, n)-multiparty schemes, whereas n refers to the number of trustees the secret is distributed among.

Shamir's secret sharing scheme assumes a polynomial f of degree $t - 1$, whereas each trustee T_i is assigned a share $d_i = f(i)$ of the secret $d = f(0) \in \mathbb{F}$, where $\mathbb{F}$ denotes an

algebraic field. By having a set $\mathcal{T}_{rec} \subseteq \mathcal{T}$ of at least t trustees assemble their shares, the secret can be reconstruced through Langrange interpolation as $f(0) = \sum_{T_i \in \mathcal{T}_{rec}} f(i) \cdot \lambda_{T_i,\mathcal{T}_{rec}}$, where $\lambda_{T_i,\mathcal{T}_{rec}} = \prod_{T_j \in \mathcal{T}_{rec} \setminus \{T_i\}} \frac{j}{j-i}$. Assuming random values $d_i \in \mathbb{Z}_q$, it is assured that knowing $t-1$ or less shares yields no advantage at guessing d.

As shown in [20], Shamir's secret sharing scheme can nicely be employed to ensure that the ElGamal secret key d used for decrypting votes is never reconstructed at any time throughout the protocol. In a first step, the trustees construct their shares d_i of private key d and the corresponding public key e in a distributed key generation protocol. As shown in [33] and [68] this can be done through mere interaction between the trustees, i.e. without requiring any given trusted party. First, each trustee T_i selects random coefficients from $\mathbb{Z}_q$ to form a polynomial $f_i(x) = \sum_{k=0}^{t-1} a_{i,k} \cdot x^k$. The basic idea of the protocol is to construct the values d_i related to an implicit $f(x)$ (a polynomial which is never constructed), where $f(x) = \sum_{T_i \in \mathcal{T}} f_i(x)$. T_i publicly commits to each coefficient by broadcasting $A_{i,0} = g^{a_{i,0}}, A_{i,1} = g^{a_{i,1}}, ..., A_{i,t-1} = g^{a_{i,t-1}}$ on $\mathcal{PB}$ and privately sends $f_i(j)$ to each other trustee T_j. For each T_i, T_j checks the received value $f_i(j)$ against the commitments by using the relation $g^{f_i(j)} = \prod_{k=0}^{t-1} (A_{i,k})^{j^k}$. If the relation does not hold, T_j makes a claim and faulty parties are excluded according to a given set of rules. Assuming for simplification that no trustees are excluded, each T_j computes his share d_j of private key d as $\sum_{T_i \in \mathcal{T}} f_i(j)$. The public key e is obtained as $\prod_{T_i} A_{i,0}$.

Example 2.7.1 *In order to ensure verifiability while preserving secrecy, the responsibilities of T_1 are distributed among a set of trustees $\mathcal{T} = \{T_1, .., T_{N_T}\}$. They generate the public key e according to the protocol outlined above and post the public information (i.e. their commitments and all resulting public values related to the ElGamal PKI) to $\mathcal{PB}$. Now in order to decrypt the product of valid votes $E(v) = (x, y) = (g^k, v \cdot e^k)$, each $T_i \in \mathcal{T}_{rec}$ computes $X_i = x^{d_i}$ and posts it to $\mathcal{PB}$ along with the following proof:*

$$ZKP[(\omega) : (X_i = x^\omega \wedge e_i = g^\omega)] \tag{2.3}$$

, where $e_i = \prod_{T_j \in \mathcal{T}} \prod_{k=0}^{N_T-1} (A_{j,k})^{i^k}$. Now anyone can compute X as $\prod_{T_i \in \mathcal{T}} X_i^{\lambda_{T_i,\mathcal{T}}}$ and the outcome of the vote.

For the sake of the simplicity the example above assumes that all trustees behave correctly, i.e. none of the trustees are actively corrupted. This yields it obsolete to mention the set $\mathcal{T}_{rec} \subseteq \mathcal{T}$. Furthermore, choosing an (n, n)-multiparty scheme, i.e. choosing higher secrecy guarantees at the cost of robustness, may in some cases do a good service to the implementation in practice. While breaking secrecy gets increasingly unlikely as n grows, with the protocol outlined above, accuracy can not be violated unnoticed. The only risk that arises when choosing t close to n is that no result can be

published, due to at least one trustee not following the protocol. In practice, the trustees should in any case be instantiated by entities that have a reputation to loose and are considered trustworthy by many. Furthermore, a cryptographic protocol alone does not necessarily answer the question of how the trustees should be implemented. Possibly a whole organisation might play the role of a trustee and it is in its own resonsibility to deliver the correct computations. In particular, it is up to the organisation to grant for the availability of its private values, for instance by keeping redundant copies.

Regarding the special case where $t = n$, we note that a far simpler protocol could be used than the one described in the example, particularly one that does not involve any private interaction among the trustees: At key-generation, each trustee T_i selects his random share d_i of secret key d. He commits to d_i by posting his share of the public key $e_i = g^{d_i}$ to $\mathcal{PB}$. Anyone can compute the public-key e as $\prod_{T_i \in \mathcal{T}} e_i$. At decryption, each trustee obtains X_i just as in the example above. Anyone can compute X as $\prod_{T_i \in \mathcal{T}} X_i$.

Now the proposed protocol in the example complies with all security requirements from section 2.2, except for coercion-resistance. Yet, we still have another problem: efficiency.

2.8 Verifiable Mix-Nets

Note that the example protocol still assumes a referendum that allows no more but two answers (yes and no) to one single question. Unfortunately, it is not easy to efficiently generalize the protocol for more complex ballots, such as elections, particularly elections that allow *write-ins* that need to be interpreted by humans. The efficiency problem concerns the voters, whose $\mathcal{OR}$-proof, which they need to cast in order to justify well-formedness, grows in the number of questions *and* the number of voting options. Although from a theoretical point of view this is not a problem when considering how we defined the voters (as having polytime computational capacities), the problem does become substantial when the voters are implemented an abstaction layer further down, i.e. in general as a person with his computer. Even when delegating the expensive computations to the computer, the performance of the current technology consitutes a real boundary to the permissible size of a potential ballot. The present section shows how to mitigate or even overcome this problem by introducing a mix-net.

The work in [20] proposes a generalization based on ElGamal and [37] improves efficiency, however in settings with a large set of possible voting options, computing the zero-knowledge proofs still takes inacceptably long on the voter's side. Thus, a practical application of performing tallying in the ciphertext space remains infeasible in many cases with ElGamal. As an alternative, privacy can be enforced by applying a mix-net which outputs re-encrypted anonymous votes for decryption, that are unlinkable to the voters they were cast by. Afterwards the votes returned by the mix-net are decrypted individually. This technique relies on the homomorphic and IND – CPA properties of ElGamal, just as the tallying procedure outlined in our example protocol. We do not revisit our example protocol, since the idea is meant to become clear from the introduction

in this section.

The foundations for mix-nets have been made by Chaum in 1981 [13]. The basic idea behind mix-nets is rather intuitive. At some point the votes need to be detached from the identities of the voters they were cast by. For exemplifying a solution to this problem, we set our focus on verifiable re-encryption mix-nets, however, also decryption mix-nets can be used, which follow the same principle. Let $\{E(v_1^x), .., E(v_{n_x}^x)\}$ denote the set of encrypted votes, which are still associated with the identities of the voters in some way. The goal is to obtain a new set $\{\hat{E}(v_{\pi(1))}^x, .., \hat{E}(v_{\pi(n_x)}^x)\}$, where π denotes a uniformly distributed random permutation from the set of all permutations over $[1, ..., n_x]$. Further, $\hat{E}(v_{\pi(i)}^x)$ denotes a re-encryption of $E(v_i^x)$, i.e. $\hat{E}(v_{\pi(i)}^x, k) = E(v_i^x, k + \hat{k})$ for a random $\hat{k} \in \mathbb{Z}_q$. Clearly, an NIZKP is required to provide verifiability.

To this end we present an intuitive but inefficient solution. The collection of votes $\{E(v_1^x), .., E(v_{n_x}^x)\}$ is sent to a first mixing node $\mathcal{M}_1$. The mixing node selects a random permutation π_1 from the set of all permutations over $[1, ..., n_x]$ and for each vote $E(v_i^x)$ a random value $k_{1,i} \in \mathbb{Z}_q$. Then $\mathcal{M}_1$ outputs $\{E(v_{\pi_1(1)}^x) \cdot E(1, k_{1,1}), .., E(v_{\pi_1(n_x)}^x) \cdot E(1, k_{1,n_x})\}$, i.e. a shuffle of each encryption mutiplied (component-wise) with an encryption of 1. The second mixing node $\mathcal{M}_2$ performs the same operations and outputs $\{E(v_{\pi_2 \circ \pi_1(1)}^x) \cdot E(1, k_{2,1}), .., E(v_{\pi_2 \circ \pi_1(n_x)}^x) \cdot E(1, k_{2,n_x})\}$. A mix-net includes at least two mixing nodes. In case of more than two mixing nodes, each subsequent one takes as its input the output of the prior one. Clearly, the outputs pertain the plaintexts (due to the homomorphic property the plaintexts are *multiplied by one*), while revealing no information on their permutation (due to the $\mathrm{IND} - \mathrm{CPA}$ property of ElGamal). In order to achieve verifiability, each mixing-node $\mathcal{M}_j$ posts its output to $\mathcal{PB}$ along with a zero-knowledge proof to show that each input is part of the output. If there is at least one honest mixing-node, vote-privacy is met.[7] The following is an intuitive but inefficient NIZKP from each mixing node $\mathcal{M}_j$:

$$ZKP[(\omega_{j,1}, ..., \omega_{j,n_x}) : \bigwedge_{i=1}^{n_x} \bigvee_{\hat{i}=1}^{n_x} E(1, \omega_{j,\hat{i}}) = \frac{\hat{E}(v_{\hat{i}})}{E(v_i)}] \tag{2.4}$$

As seen in section 2.5, proving the equality of discrete logarithms costs the prover two exponentiations. The expense of that is scaled by the square of the number of cast votes in example 2.4. Clearly, an application in practice is infeasible for a large-scale setting. As summarized in [41], several proposals have been developed to provide efficiency and satisfy even large-scale elections [32], [62], [89] and [38]. According to [41], the best proposals require only between $6 \cdot n_x$ and $8 \cdot n_x$ modular exponentiations for generating the proof and between $6 \cdot n_x$ and $10 \cdot n_x$ modular exponentiations for proof verification. A separate class of techniques to generate the proofs is called *randomized partial checking* (RPC), which provides even better efficiency, however, at the cost of

[7] Strictly speaking, one may argue that there must be at least two honest mixing nodes in order achieve secrecy - consider the case where all other mixing nodes reveal their permutation to the honest one. However, since the honest mixing node is not corrupted, it will not misuse the information. Further, note that in this simple example the randomness strictly needs to be unknown to the mixing nodes in order to avoid mapping multiple ciphertexts to the same element of the output.

soundness. The original work already relates RPC to voting [47]. Recently new attacks have been published against RPC in [52] and attacks against another prominent mix-net that holds RPC elements are published in [51].

Finally, chapter 3 shows possible solutions for achieving coercion-resistance and - as a necessary condition - anonymity. Note, that the solution in the example indeed allows voters to prove to a coercer how they voted, simply by issuing him the employed randomness k. The coercer can easily verify whether the received randomness is correct (ElGamal x-component) and use it to decrypt the vote (ElGamal y-component). Even easier, the coercer could tell the voter to hand out his voting credential and use it to cast a vote himself.

2.9 Conclusion

In the course of the previous sections we have introduced and refined an example voting protocol. By applying the introduced building blocks, we managed to satisfy the requirements *democracy*, *accuracy*, *secrecy* and *verifiability* from section 2.2 under the trust assumptions from section 2.3. However, it would be wrong to claim the trustworthiness of a voting system simply by arguing that the presented protocol is implemented. We need to keep in mind that we have considered voters, trustees and attackers on a very high level of abstraction - they are essentially modeled as machines with a free will or, vice-versa, as persons that think as fast as machines. This level of abstraction is chosen in many protocol descriptions throughout the literature, moreover in most of the ones introduced in the following chapters. We may not neglect that designers will need to instantiate the voters, trustees and the attacker in the protocol description from the top down to the implementation details. Unappropriate decisions may render the addded value of any good protocol useless.

As an example, it seems natural to imagine the voters from the protocol as pairs of humans and their home computers. In general, home computers are not considered trustwothy nowadays. At the same time, one may think of hostile origanisations trying to manipulate a result by exploiting the insecurity of the voters' home computers. Such concerns may particularly arise in the context of large-scale political elections. Protocol descriptions that do not model the voters' computers, inherently fail to capture and reduce risks of this kind, i.e. risks that may be perceived as real and actually obvious in certain settings. In such a case, the added value of a protocol does not only fail at increasing security, but it also suffers in its function of explaining how and to which degree major risks are reduced. The Norwegian project for the 2011 political elections developped and ran a system that started from a model that *does* explicit the voters' computers as separate system players [35]. Interestingly, their model had a great impact on the protocol definition, which in return had a great impact on the implementation and the ability to explain and debate to which degree it may be considered trustworthy. Many protocols that have been known in the literature for a long time, would have been unsuitable for this purpose. Another protocol that considers the problem of home computers being insecure is used in the scheme called *pretty good democracy* [43].

Just as with the voters, protocol descriptions generally do not specify how *the trustees* should be implemented. This accounts for the trustees identified in the example protocol, just as much as for the ones implied by the instantiated building blocks, as in mix-nets or public bulletin boards. In secure multiparty computation, mix-nets or public bulletin boards, the aim of proposing a group of trustees - rather than just one - lies in reducing the risk inherent to having a single, possibly fraudulent party. Clearly, the ambition must lie in instantiating a sufficient number of trustees that are perceived as trustworthy and independent from eachother. After all, the trustworthiness of the overall system will be assessed according to their implementation, not just the underlying protocol.

We conclude that trustworthy Internet voting is not achieved by just implementing a provably secure protocol. However, a good protocol allows to take advantage of the trustworthiness of the implementing components and bring that trustworthiness to its best. The proposed building blocks support the definition of such protocols.

Chapter 3

Towards Efficiently Combining Verifiability and Coercion-Resistance

This chapter contains the results from three peer-reviewed publications [82], [83] and [74]. The first two were written under strong participation of the author of this thesis. In a few cases, some statements may be taken from these papers with only minor changes.

In 2005, Juels, Catalano, and Jakobsson [49] have proposed a scheme that inspired the work of many researchers in the years that followed. It provides a very strong sense of verifiability and coercion-resistance, both at the same time. Still today it is often referred to as *the* JCJ protocol. The authors perceive receipt-freeness as a mere precondition to coercion-resistance. Clearly, the voters should not be able to demonstrate to adversaries how they voted. But the JCJ scheme goes even further. The protocol also protects against adversaries who try to obtain the voters' credentials (simulation attack), keeping them from casting a vote (forced abstention attack) or forcing them into voting at random (randomization attack). They propose to consider protocols *coercion-resistant* only if they manage to render all of these attacks impossible, under the weak trust assumptions introduced in the previous chapter.

In the context of this thesis, we generally do not distinguish between coercers (people who punish) and vote-buyers (people who reward). We rather focus on the question whether an adversary can distinguish if a voter follows his instructions or if he applies a counter-strategy. However, we do address this difference in the context of related work in section 3.5.2. For the other parts, refer to section 2.2 for an intuitive definition of the term coercion-resistance.

The JCJ scheme generally renders the phase of tallying unbearably long and it can not be employed for real ballots. Particularly, the running-time is square in the number of voters. Nevertheless, the protocol is widely discussed and taken as a starting point for further improvements. The publications [82] and [83] have been proposed under strong participation of the author of this thesis. They are presented in-depth, each in a designated section. Both proposals aim at increasing the efficiency of the tallying procedure in JCJ.

In section 3.1 we provide some background for assessing coerion-resistant schemes.

After presenting the JCJ protocol in section 3.2, we present our alternatives in sections 3.3 and 3.4. Section 3.5 presents proposals which are related to our schemes. Finally, we conclude the chapter in section 3.6.

3.1 Prerequisites

This section outlines the prerequisites that are shared by many proposals in coercion-resistant Internet voting. Section 3.1.1 shows three building blocks that are used within a number of schemes. They have not been outlined previously in chapter 2, since they are rather specific to the present domain. However, the building-blocks introduced in the previous chapter are also relevant here. Section 3.1.2 shows the assumptions concerning the behaviour of the players and the adversary. Then, in section 3.1.3 we argue why coercion-resistance cannot be provided unconditionally. In the extreme case where the coercer can predict the behaviour of each single voter (apart from the one he wants to coerce), coercion will always be possible. A coercive attacker inherently always needs to face some degree of noise in the voters' behaviour. (Recall the notion of *adversarial uncertainty* introduced in section 2.3.) Based on these reflections, section 3.1.3 also introduces an intuitive measure for the degree of coecrion-resistance (δ). Finally, section 3.1.4 shows how the coercion-resistance of protocols can be assessed based on an appropriate model, i.e. that uses the measure δ.

3.1.1 Additional Particular Building Blocks

The example protocol from chapter 2 is clearly vulnerable to all of the coercion types introduced above. As an example, a coercer may furnish a voter with the ciphertext to post (he can verify the voter's compliance by finding the legitimate vote on $\mathcal{PB}$, i.e. the receipt). He can also ask the voter to hand out his voting credential and vote by himself (simulation attack) or tell the voter not to vote and observe $\mathcal{PB}$ to find no vote associated with the voter's public key (forced abstention attack). In order to overcome these issues, the literature typically proposes the following four additional building-blocks.

Untappable Channels

In JCJ, the voters obtain their voting credential through an untappable channel. This enables them to lie about their credential. Due to untappable channels, coercers will not know any of the passed information, not even the ciphertexts. An effective implementation might entail having the voters visit a trusted envirnoment.

Anonymous Channels

By eavesdropping the channel to $\mathcal{PB}$, the coercer can always find out whether the voter cast a vote. In coercion-resistant Internet voting it is crucial, that voters are not continuously observed and that they can cast their vote through a channel that is not corrupted

by the coercer. Anonymous channels can be implemented by mix-nets that serve multiple peers. Thus, the coercer can only witness the voter communicating with $\mathcal{PB}$ in the event where many mixing-nodes are corrupted. *Tor* is a renowned implementation of an anonymous channel [24].

Plaintext Equality Test - PET

Even in the presence of an anonymous channel, the use of a signature infrastructure still offers simple ways of performing all four kinds of coercion attacks. By observing $\mathcal{PB}$ the coercer will always know whether the voter abstained as requested, or if the private key he provided as his voting credential is correct. Clearly, no information on $\mathcal{PB}$ may allow to decide whether a vote that has just been cast is authentic, neither may votes that are assessed as authentic be associated with voters' identities. A straight-forward approach is to use $\mathrm{IND}-\mathrm{CPA}$-secure homomorphic encryptions of voting credentials. Thus, on $\mathcal{PB}$ there are ElGamal encryptions of voting credentials associated with the voters' identities. Along with their vote, the voters post another encryption of the same voting credential to $\mathcal{PB}$. This is the approach chosen in JCJ. Clearly, at some point in the protocol - after applying mix-nets for anonymization - ciphertexts will need to be compared for authentication, i.e. the ciphertexts that emerged from the ones originally on $\mathcal{PB}$ (we call them $\hat{S}$), with the ones that emerge from vote casting (we call them $\hat{A}$). Decrypting them is not an option, since that would reveal voting credentials and allow coercion. Plaintext equality tests (PET) resolve this problem by allowing to test whether two given ciphertexts encrypt the same plaintext without needing to decrypt them. Remarkably, this can easily be done in a verifiable manner (PET provides an NIZKP to justify the assessment of the plaintexts' equality) and without letting the verifiers of the NIZKP learn anything about the plaintexts with more than negligible probability. PET can be performed in a distributed setting, i.e. by multiple trustees, whereas only a majority can learn the plaintexts. Let E_1 and E_2 denote the two ciphertexts. Apparently the decryption of $(\frac{E_1}{E_2})^z$ for random $z \in \mathbb{Z}_q$ equals 1 if and only if they are encryptions of the same plaintext. PET has been proposed in [46].

Modified Plaintext Equality Test - M-PET

As mentioned in the previous paragraph, at some stage in the JCJ-protocol, plaintext equality tests need to be performed in order to authenticate. Since the encrypted voting credentials need to be mixed and re-encrypted beforehand, the talliers lack any heuristic of how to proceed efficiently. In effect, they need to perform PET on all combinations of elements from $\hat{S}$, a set holding N elements, and $\hat{A}$, a set holding up to n_{cast} elements. Note, that coercion-resistant Internet voting needs to allow voters to cast any number of unauthentic votes to $\mathcal{PB}$, otherwise coercers will always succeed with simulation attacks. Therefore, $n_{cast} - n_{authentic}$ may even be greater than N. This is part of the reason why tallying is inefficient with JCJ. Although PET cannot be simply be replaced by M-PET as proposed in [88], some protocols offer coercion-resistance by instantiating M-PET at some point. Given ciphertexts $E_(x_1), ...E_(x_n)$, M-PET raises all values to the

power of a random value $z \in \mathbb{Z}_u$, and decrypts them to obtain the blinded plaintexts $x_1^z = DEC(E(x_1)^z), ..., x_n^z = DEC(E(x_n)^z)$. The blinded plaintexts can be efficiently compared for equality for instance by sequentially saving them in a hash-table [88]. A collision is detected if and only if the plaintexts are equal. M-PET reveals no non-negligible information of the plaintexts, under the condition that the discrete logarithm of any plaintext x_i is unknown in the base of any plaintext x_j, where $1 \leq i < j \leq n$ and $x_i \neq x_j$. Just as PET, M-PET is verifiable by the means of an NIZKP and can be performed in a distributed setting.

3.1.2 Assumptions on Players and the Adversary

Schemes that aim at providing coercion-resistant Internet voting divide voters $\mathcal{V}$ into the following sub-groups: honest voters $\mathcal{V}^{\mathcal{H}}$, corrupted voters $\mathcal{V}^{\mathcal{A}}$ and one voter $V^{\mathcal{A}_C} \notin \mathcal{V}^{\mathcal{H}} \cup \mathcal{V}^{\mathcal{A}}$ who is subject to coercion by a static and active adversary $\mathcal{A}$[1]. Trustees are divided into the sub-groups of registrars $\mathcal{R}$ and talliers $\mathcal{T}$[2]. $\mathcal{A}$ can corrupt $\mathcal{T}^{\mathcal{A}}$, which is a minority of $\mathcal{T}$, and all members of $\mathcal{V}^{\mathcal{A}}$. His goal is to coerce $V^{\mathcal{A}_C}$ by controlling the behaviour of all corrupted players. He may also want to violate against any other requirement identified in section 2.2. In this case, we apply the assumptions identified in section 2.3.

A coercion-resistant protocol needs to furnish $V^{\mathcal{A}_C}$ with a counter-strategy, such that he may credibly claim having followed $\mathcal{A}$'s orders, even when he has not. We do not necessarily asume that $V^{\mathcal{A}_C}$ actually wants to take advantage of the counter-strategy. Yet, the counter-strategy is needed in order for $\mathcal{A}$ to be uncertain regarding the voter's behaviour.

Clearly, when voters register, none of the members of $\mathcal{R}$ may be corrupted - one corrupted registrar would be sufficient to mount a forced-abstention attack. Therefore we need to assume that there is no coercion during registration. However, during all other phases of the protocol we may assume a minority $\mathcal{R}^{\mathcal{A}} \subset \mathcal{R}$ to be corrupted.[3] For $V^{\mathcal{A}_C}$ to be protected against corrupted registars, in addition we need to assume that *he knows* the minority of trustworthy registrars, i.e. from $\mathcal{R} \setminus \mathcal{R}^{\mathcal{A}}$. We will get to that point later in section 3.2.1.

[1] Unlike adaptive adversaries, static adversaries choose which players to corrupt prior to protocol execution. Active adversaries take full control over the players they corrupt, whereas passive adversaries would only obtain the corrupted players' knowledge.

[2] In this chapter, $\mathcal{T}$ does not denote the full set of trustees.

[3] We recall from section 2.7 the possibility of operating an (n, n)-multiparty scheme. If this is applied one can go as far as assuming only one single trustworthy member of $\mathcal{R}$. The same accounts for $\mathcal{T}$. This may not be recommended due to robustness concerns. Yet, in section 2.7, we argue why it can be reasonable to shift the threshold close to n.

3.1.3 Adversarial Uncertainty and a Measure for the Degree of Coercion-Resistance

In section 2.3 we argue why we need to be able to assume adversarial uncertainty regarding the final tally Σ and the number of unauthentic votes Γ, which are outruled in the course of tallying. Clearly, the behaviour of voters from $\mathcal{V}^{\mathcal{H}}$ needs to be sufficiently unpredictable and diversified. Adversarial uncertainty is supported by a sufficiently large set $\mathcal{V}^{\mathcal{H}}$ in combination with a sufficiently small set $\mathcal{C}$ of voting options.

In [57], Küsters et al. introduce a measure for quantifying the degree of coercion-resistance. As an example, they assume an adversary who knows the probability for each voting option to be chosen, i.e. the probability of $v_i^{valid} = c_j$, for all $1 \leq i \leq N$ and all $1 \leq j \leq N_c$. These probabilities are equal for all voters from $\mathcal{V}^{\mathcal{H}}$ who cast a valid vote. By applying stochastics on his prior knowledge and Σ, the adversary decides, whether it is more probable that $V^{\mathcal{A}_C}$ applied his defense strategy and cast the ballot of his preference or not. Clearly, this is the best way to distinguish whether $V^{\mathcal{A}_C}$ complied or not and therefore the best incentive for $V^{\mathcal{A}_C}$ to actually comply. In case the stochastics imply that the voter acted according to his free will, the adversary will choose not to reward $V^{\mathcal{A}_C}$.

The degree of coercion-resistance δ is defined as the probability that a reasonable adversary will accept a run (and thus reward $V^{\mathcal{A}_C}$), given that $V^{\mathcal{A}_C}$ submits to coercion, minus the probability that the adversary will accept a run, given that $V^{\mathcal{A}_C}$ applies the defense strategy. More concisely, $\delta = P(A|B) - P(A|\neg B)$, where A denotes the event *adversary accepts run* and B denotes $V^{\mathcal{A}_C}$ *submits to coercion*.

If the adversary offers a voter 100 dollars for a vote when using a system that allows no defense strategy, the voter may expect to get the full reward when submitting to coercion and nothing otherwise. If there is a defense strategy, δ signifies the fraction of the 100 dollars voters may in average expect to additionally get from a vote buyer when submitting to coercion, as opposed to applying a defense strategy. Obviously, the smaller δ, the higher the resistance against coercion.

Example 3.1.1 *Assume an adversary knows the distribution of* Γ*, given that* $V^{\mathcal{A}_C}$ *abstains from voting. Let* Γ *be uniformly distributed over one to ten. Based on this knowledge and by observing* $\mathcal{PB}$*, he mounts a forced abstention attack and offers* $V^{\mathcal{A}_C}$ *50 dollars for not casting any vote. If* $V^{\mathcal{A}_C}$ *abstains, he may expect to get the money with a probability of 100%. If he does cast a vote, the risk is 10% that* Γ *will take the value 11, i.e. the only case where the stochastics imply that* $V^{\mathcal{A}_C}$ *did not comply. In this case* $V^{\mathcal{A}_C}$ *will only get the money with 90% probability. Therefore* $\delta = \frac{1}{10}$ *and on average the voter may expect 5 dollars more when he decides to comply rather than casting his vote.*

Note that coercion based on Σ is not specific to Internet voting. Γ however might be. On one hand, since coercion-resistant schemes that employ $\mathcal{PB}$ are not in practice yet, adversarial uncertainty with regard to Γ is to be expected in real life. On the other hand, since also voters are uncertain about Γ, the adversary can still launch an attack that grounds on a wild guess $\Gamma = a$: He can offer money in case $\Gamma \leq a$ or scratch $V^{\mathcal{A}_C}$'s

car if $\Gamma > a$. The reasonable voter will submit to coercion if he believes that the vote cast with the fake credential would cause Γ to exceed a by 1. Since in a scheme that is meant to be coercion-resistant there is no reason to actually post illegitimate votes to $\mathcal{PB}$, a might initially be chosen relatively small, thus yielding δ correspondingly high.

3.1.4 Assessing Coercion-Resistance

The original JCJ-paper [49] proposes a model to capture the authors' notion of coercion-resistance. Based on that, they prove that their scheme is coercion-resistant ($\delta = 0$), given the assumptions introduced in 3.1.2 and adversarial uncertainty regarding Σ and Γ. Here we provide a simplified exposition of the model, thus focusing on the important aspects.

The model proposed by the authors of the JCJ scheme assumes an adversary $\mathcal{A}$ that tells voter $V^{\mathcal{A}_C}$ to hand out his voting credential **cred.** $\mathcal{A}$ can use the received credential to cast a vote. By controlling the actions of the corrupted players and by observing $\mathcal{PB}$, he tries to find out, whether $V^{\mathcal{A}_C}$ handed out the correct voting credential or a fake one **cred**$'$. If his guess is right, he wins. We describe this challenge as the *real coercion game* in listing 3. It is assumed that a simulator $\mathcal{S}$ runs the vote by performing the *real* protocol using the building-blocks introduced in chapter 2 and section 3.1.1, i.e. indistiguishably from a real vote.

Algorithm 4 Real coercion game

Require: $\mathcal{V}$, $\mathcal{C}$ // identities of all voters, i.e. $\mathcal{V} = \mathcal{V}^{\mathcal{H}} \cup \mathcal{V}^{\mathcal{A}} \cup \{V^{\mathcal{A}_C}\}$

1: $\mathcal{V}^{\mathcal{A}} \leftarrow \mathcal{A}$ selects who to corrupt
2: $\mathcal{PB} \Leftarrow$ Run Registration of all $V \in \mathcal{V}$
3: $(V^{\mathcal{A}_C}, v_{\mathcal{A}} \in \mathcal{C}) \leftarrow \mathcal{A}$ selects who to coerce and how he should vote in his defense strategy
4: $b \leftarrow$ uniformly random from $\{0, 1\}$
5: **if** $b = 0$ **then**
6: Give $\mathcal{A}$ **cred**$'$
7: $\mathcal{PB} \Leftarrow$ Cast vote $v_{\mathcal{A}}$ using **cred**
8: **else**
9: Give $\mathcal{A}$ **cred**
10: **end if**
11: $\mathcal{PB} \Leftarrow$ Post votes of $\mathcal{V}^{\mathcal{H}}$ based on prior knowledge on Σ
12: $\mathcal{PB} \Leftarrow \mathcal{A}$ posts ballots of $\mathcal{V}^{\mathcal{A}}$ and a vote using **cred** or **cred**$'$, depending on b
13: $\mathcal{PB} \Leftarrow$ Perform tallying and produce Σ
14: **return** $\mathcal{A}$'s guess on b, given $\mathcal{PB}$

Note, that in the model it is $\mathcal{A}$ who chooses the vote $v_{\mathcal{A}}$ for $V^{\mathcal{A}_C}$ to cast within his defense strategy. Although this is counter-intuitive, it gives the attacker a lot of power - if $\mathcal{A}$ cannot distinguish $V^{\mathcal{A}_C}$'s behaviour under this condition, he will definitely not be able to do so if $V^{\mathcal{A}_C}$ gets to choose his vote on his own. Further, we note that $\mathcal{A}$ might

want the voter to vote (for $v_{\mathcal{A}}$) while the voter has no vote to cast and wants to abstain. This case is captured by assuming the option *abstain* as an element of the voting options $\mathcal{C}$. Voters V_i can cast a vote with their fake credential $\mathbf{cred}'_i$ eventhough they are not under coercion. This is captured in the game by allowing *cast vote with fake credential* as a voting option in $\mathcal{C}$.

In addition to the real coercion game, JCJ proposes the notion of an *ideal coercion game*. Again, $\mathcal{A}$ is supposed to guess the value of b. However this time, he will always get $V^{\mathcal{A}_C}$'s real credential **cred**, regardless of b. Further, $\mathcal{A}$ needs to make his guess solely based on Σ. He has no access to $\mathcal{PB}$. The ideal coercion game serves as a benchmark for the notion of ideality regarding coercion-resistance. It is shown in listing 5. Note, that the vote posted on behalf of $V^{\mathcal{A}_C}$ (line 10) may not be counted in the final tally.

Algorithm 5 Ideal coercion game

Require: $\mathcal{V}$, $\mathcal{C}$ // identities of all voters, i.e. $\mathcal{V} = \mathcal{V}^{\mathcal{H}} \cup \mathcal{V}^{\mathcal{A}} \cup \{V^{\mathcal{A}_C}\}$

1: $\mathcal{V}^{\mathcal{A}} \leftarrow \mathcal{A}$ selects who to corrupt
2: $\mathcal{PB} \Leftarrow$ Run Registration of all $V \in \mathcal{V}$
3: $(V^{\mathcal{A}_C},\ v_{\mathcal{A}} \in \mathcal{C}) \leftarrow \mathcal{A}$ selects who to coerce and how he should vote in his defense strategy
4: $b \leftarrow$ uniformly random from $\{0, 1\}$
5: **if** $b = 0$ **then**
6: $\quad \mathcal{PB} \Leftarrow$ Cast vote $v_{\mathcal{A}}$ using **cred**
7: **end if**
8: Give $\mathcal{A}$ **cred**
9: $\mathcal{PB} \Leftarrow$ Post votes of $\mathcal{V}^{\mathcal{H}}$ based on prior knowledge on Σ
10: $\mathcal{PB} \Leftarrow \mathcal{A}$ posts ballots of $\mathcal{V}^{\mathcal{A}}$ and a vote using **cred**
11: $\mathcal{PB} \Leftarrow$ Perform tallying and produce Σ
12: **return** $\mathcal{A}$'s guess on b, given Σ

Now we can give the following definition of coercion-resistance: *A scheme is coercion-resistant, if $\mathcal{A}$ has only negligible advantage at winning the real coercion game over winning the ideal coercion game, in some security parameter.*

Note that apart from Σ, $\mathcal{A}$ will learn Γ not just in the real game but also in the ideal game.[4] Further, in the previous section we have observed that δ emerging from prior knowledge on Σ and Γ, is likely to be *very small* in large settings, however not *negligible* in the sense of the common definitions. In other words, the ideal game is not that ideal after all. Unfortunately it is not possible to quantify this deficiency using the measure δ introduced in the previous section - clearly, the value of δ related to Σ and Γ solely depends on the context of the vote, not at all on the employed voting scheme. However, this observation gives some justification for considering schemes that are not entirely

[4] Actually he learns just the number of *all* invalid votes that are ruled out during tallying. However, he does not know which ones were ruled out due to being spoiled, duplicate or - the actual Γ - unauthentic. We do not further discuss this detail.

coercion-resistant in the sense of the present definition. The schemes we propose in sections 3.3 and 3.4 are indeed δ-coercion-resistant, whereas $\delta > 0$ is not sensitive to the unknown values Σ and Γ, but to a parameter β. Clearly, we need to adapt the present model in order for it to capture δ-coercion-resistance. With the ambition of staying as close to the JCJ-model as possible, we do not change the coercion games and give the following definition for δ-coercion-resistance instead: *A scheme is δ-coercion-resistant, if $\mathcal{A}$ has an advantage of at most negligibly more than δ at winning the real coercion game over winning the ideal coercion game, in some security parameter.*

3.2 Protocol by Juels et al. 2005 (JCJ)

To achieve receipt-freeness, which is a strict necessary condition for coercion-resistance, other protocols need to assume an *untappable channel* [71] between authorities and voters at every voting event. Requiring voters to visit the authorities' offices at each occasion clearly compromises the spirit of remote voting. JCJ is distinguished by assuming an untappable channel only during the distribution of the voters' credentials. Since JCJ allows credentials to be re-used in many subsequent voting events, they can be distributed when citizens appear in person at the administration offices to register as new community members. We start off by introducing the protocol in 3.2.1. In section 3.2.2 we briefly argue in an informal way, to which degree JCJ satisfies verifiability. Using the model introduced in section 3.1.4, coercion-resistance (and thus vote-privacy and anonymity) will be demonstrated more extensively in section 3.2.3. Fairness is evident and left undiscussed.

3.2.1 Description of the Protocol

In the following paragraphs, we present each phase of the JCJ protocol. To keep our exposition consise, we refer to the primitives presented in chapter 2 and the previous section. We assume the application of publicly verifiable multiparty computation whenever registars or talliers perform joint computations. All ciphertexts are ElGamal encryptions over a pre-established multiplicative cyclic group $(\mathbb{G}_q, \cdot, 1)$ of order q, for which the decisional Diffie–Hellman problem (DDH) is assumed to be hard.[5] A public key infrastructure is in place, where the talliers $\mathcal{T}$ share the private key d.

[5]We thus follow Civitas [18], which basically instantiates the JCJ protocol. However, they do deviate in the choice of the underlying cryptosystem. The reason behind JCJ choosing a modified version of ElGamal (M-ElGamal) lies in the reasoning in their security proof. Although we could allow our protocol to adopt M-ElGamal as well, we adhere to the more standard ElGamal, thus making its performance more easily comparable to most of the other known proposals for coercion-resistant Internet voting. Further, the question whether to choose ElGamal or M-ElGamal does not seem sensitive to the design of a particular verifiable voting protocol, but rather to the desired security reassurances of the cryptosystem itself. Notably, recently ElGamal has been proved to have the beneficial IND-CCA1 property (resistance against non-adaptive chosen ciphertext attacks) just as much as M-ElGamal [60].

Registration. The registrars $\mathcal{R}$ add voter V to the set of eligible voters $\mathcal{V}$. They jointly compute the random value $\sigma \in \mathbb{G}_q$ and keep their shares to themselves. V's voting credential **cred** is exactly σ. They compute its public counterpart **Cred** as $E(\sigma)$. They append **Cred** to the list IDENTIFIABLECREDENTIALS on $\mathcal{PB}$ and associate it with an identifier of V, such as name, birthday and address. In JCJ this list is called the *voter-roll*. They send the voter both values through an untappable channel, along with a ZKP to prove that **Cred** is an encryption of **cred**. V verifies the proof.

We now elaborate on how the voter is protected from corrupted registrars. Recall from section 3.1.2 that V needs to be able to give $\mathcal{A}$ a fake credential **cred**$'$ such that $\mathcal{A}$ cannot distinguish whether having received the real credential **cred** or not. This clearly hinges on the way the registrars establish σ, $E(\sigma)$ and the proof. Each registrar R_i from $\mathcal{R}$ encrypts his share of $\mathbf{cred_i}$ as $(x, y) = (g^{k_{R_i}}, \sigma_i \cdot e^{k_{R_i}})$.[6] We call this value $\mathbf{Cred_i}$. He sends both $\mathbf{cred_i}$ and $\mathbf{Cred_i}$ to the voter through the untappable channel. The voter computes **cred** as $\prod_{\mathcal{R}} \mathbf{cred}_i$ and verifies that the correct value is published in IDENTIFIABLECREDENTIALS by comparing **Cred** with $\prod_{\mathcal{R}} \mathbf{Cred}_i$. At this point, not knowing k_{R_i}, V cannot prove to the adversary that the credential he claims to be correct is indeed **cred**. However, so far he has no way to decide whether $\mathbf{Cred_i}$ is actually an encryption of $\mathbf{cred_i}$. To solve this dilemma, R_i provides a non-transferable, so-called *designated verifier proof* $ZKP[(\omega) : ((x = g^{\omega}) \wedge (y = \frac{e^{\omega}}{\sigma_i})) \vee \hat{e} = \hat{g}^{\omega}]$ along with both values, similarly as proposed in [45]. The values $\hat{e}$ and $\hat{g}$ are V's public key and the generator of some discrete-log based signature infrastructure. Since V is the only player who actually knows the private key $\hat{d}$ that matches $\hat{e}$, he learns from the proof that $\mathbf{Cred_i}$ is indeed an encryption of $\mathbf{cred_i}$. However, using $\hat{d}$, V is able to simulate such a proof. Thus, V can always change $\mathbf{cred_i}$ from the registrar he believes to be trustworthy throughout the protocol to $\mathbf{cred_i}'$ and compute **cred**$'$ accordingly. He gives $\mathcal{A}$ the simulated proof along with $\mathbf{cred_i}'$. As an alternative to using designated-verifier proofs, we can also use interactive ZKP, which are non-transferable by nature.

Vote Casting. V identifies his prefered vote v from the available set of valid voting options $\mathcal{C}$. He constructs the tuple $(E(\sigma), E(v), \Pi_1, \Pi_2)$ and posts it to $\mathcal{PB}$ through the anonymous channel. On $\mathcal{PB}$ it is appended to the set CASTVOTES.

Π_1 and Π_2 are two NIZKP. Π_1 is there to prove knowledge of σ and Π_2 is there to prove $v \in \mathcal{C}$, which clearly requires knowledge of c as well. Requiring Π_1 prevents attackers from casting valid votes by re-encrypting entries from IDENTIFIABLECREDENTIALS. Since each valid vote on $\mathcal{PB}$ will be decrypted during the tallying phase, Π_2 is needed to prevent $\mathcal{A}$ from forcing voters to select $v \notin \mathcal{C}$ according to some prescribed pattern. Thus he would obtain a receipt from $\mathcal{PB}$. This attack strategy is known as the *Italian attack* [22]. In more general terms, this can be seen as a measure to keep the space of the published decrypted votes small for the sake of adversarial uncertainty (refer to section 3.1.3). However, the cost is high on the voter's side, notably something we managed to avoid in the example protocol of the previous chapter. Note, that proving the mere

[6] A majority of registrars would be sufficient, we refer the reader to 3.1.2. For the sake of simplicity we prefer to assume the participation of all trustees of a given group in our further exposition.

knowledge of v is needed to prevent the system from getting misused as a decryption oracle. Otherwise, the adversary could learn valid credentials by selecting $E(v)$ as a re-encryption of an element from IDENTIFIABLECREDENTIALS and use them in subsequent votes.

Tallying. At the end of the vote casting phase, CASTVOTES contains n posted votes, generally not all of which should be counted. We divide the tallying phase into five steps. The steps are also summarized in figure 3.1.

1. **Check Π_1:** The set CASTVOTES holds elements of the shape $(E(\hat{\sigma}_i), E(\hat{v}_i), \widehat{\Pi}_{1,i}, \widehat{\Pi}_{2,i})$. The hat symbolizes a value yet to be checked. The talliers $\mathcal{T}$ verify each $\widehat{\Pi}_{1,i}$. In case of success, the element is appended to the set OWNVOTES, whereas the Π_1-component is dropped.

2. **Check Π_2:** The set OWNVOTES holds elements of the shape $(E(\hat{\sigma}_i), E(\hat{v}_i), \widehat{\Pi}_{2,i})$. This time, $\mathcal{T}$ check all $\widehat{\Pi}_{2,i}$. Upon success, the element is appended to WELLFORMEDVOTES. Again the proof is no longer needed.

3. **Remove duplicates:** The set WELLFORMEDVOTES holds elements of the shape $(E(\hat{\sigma}_i), E(v_i))$. $\mathcal{T}$ apply PET pair-wise on $E(\hat{\sigma}_i)$ and $E(\hat{\sigma}_j)$, for all $1 \leq i < j \leq n_{well-formed}$. In case of two equal plaintexts, the corresponding votes are duplicates and one needs to be dropped. Since WELLFORMEDVOTES contains the votes in the order as cast, $\mathcal{T}$ can enforce a policy on which vote to count, for instance the first or the last one. In [78] the benefits of counting the last ballot are pointed out. The votes that are not ruled out are appended to the set UNIQUEVOTES. Note, that the running time of this step is $\mathcal{O}(N^2)$, assuming that voters cast one vote on average.

4. **Remove Unauthentic Votes:** The sets UNIQUEVOTES and IDENTIFIABLECREDENTIALS are each passed to a mix-net. The output for UNIQUEVOTES is the set UNLINKABLEVOTES, which takes the shape $(E(\hat{\sigma}_i), E(v_i))$. The other output is UNLINKABLECREDENTIALS, which holds values $E(\sigma_i)$. Now $\mathcal{T}$ perform PET pair-wise on all $E(\hat{\sigma}_i)$ and $E(\sigma_j)$, where $1 \leq i \leq n_{unique}$ and $1 \leq j \leq N$. In case of equal plaintexts, $E(v_i)$ is appended to the set VALIDVOTES. This procedure runs in $\mathcal{O}(N^2)$ time, assuming that voters cast one vote on average.

5. **Decrypt and Count:** $\mathcal{T}$ decrypt all elements of VALIDVOTES, count and publish the result as Σ.

3.2.2 Verifiability

We briefly argue in an informal way, to which degree JCJ satisfies verifiability. Recall from section 2.3 that a protocol is meant to be verifiable, even if all trustees are corrupted. By observing $\mathcal{PB}$, voters verify that their vote has been cast as intended and recorded as cast. Further they know from the proof at registration that the value they used as

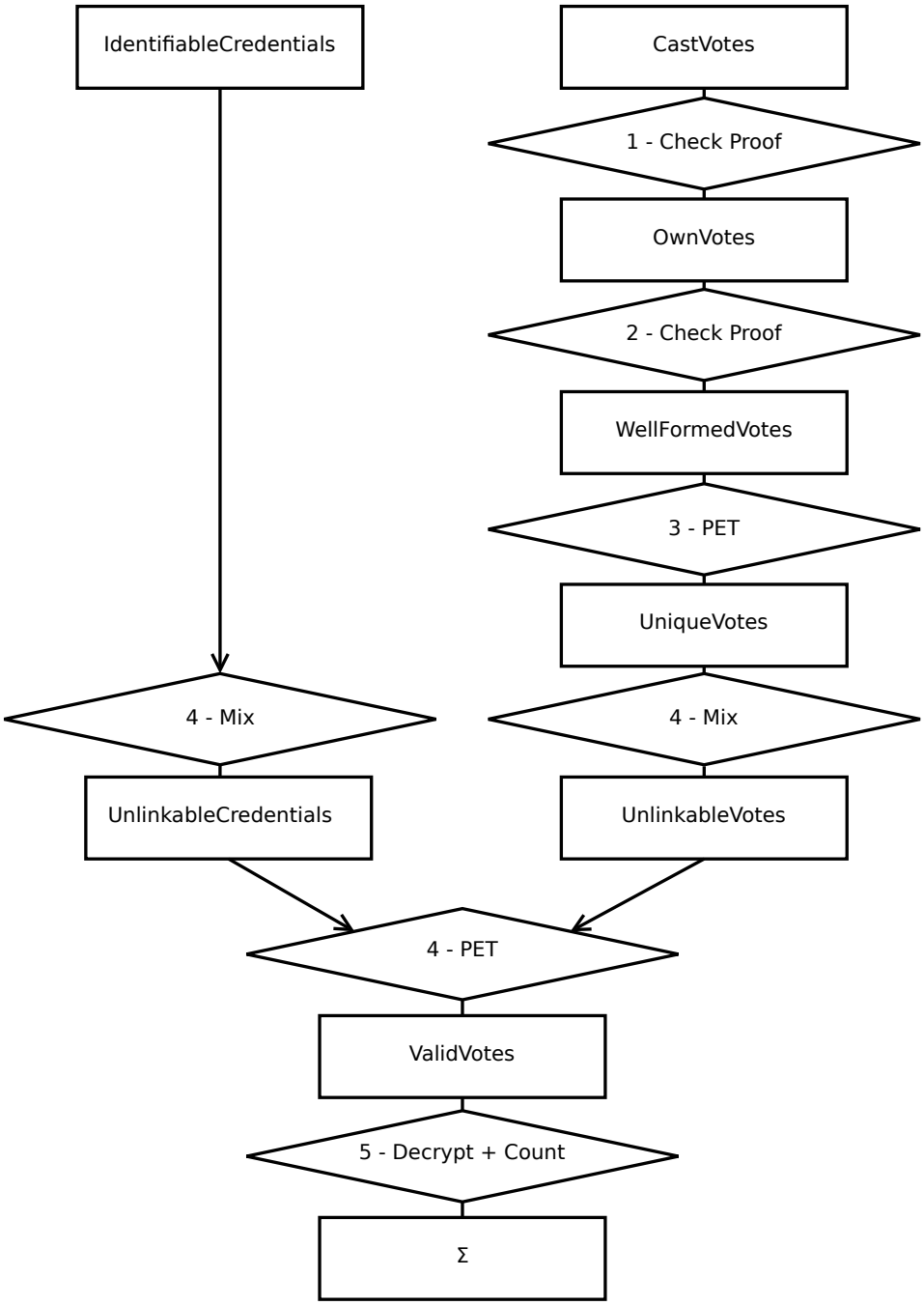


Figure 3.1: Tallying phase of the JCJ protocol.

their voting credential matches their value in IdentifiableCredentials. When also verifying the proofs generated at tallying, they know that their vote has been counted as recorded.

Clearly, a colluding majority of authorities could secretly decrypt V's entry in IdentifiableCredentials and obtain σ. However, if they use σ for casting votes, they

could be exposed by V when the corresponding PET algorithm returns *true* at removing duplicates during the tallying procedure.

Since the protocol requires votes to be assessed against values that are linked to voters' identities, the registrars cannot create any unassigned credential that would allow casting a vote. If they could, the protocol would violate against *eligibility verifiability* shortly mentioned in section 2.2. Although the original JCJ-paper mentions *verification against voter-rolls*, it does not call it a precondition for verifiability. Yet, their scheme is also verifiable even in this sense.

3.2.3 Proof-Sketch for Coercion-Resistance

Here we sketch a proof to show that JCJ is coercion-resistant based on the definition from section 3.1.4. We will re-use these arguments for the protocols in the following two sections.

The simulator $\mathcal{S}$ runs the real coercion game. He simulates the actions of $\mathcal{R}$ and $\mathcal{T}$ without deviating from the protocol. (Doing so, he gets to choose the decryption key d.) $\mathcal{A}$ cannot tell the difference on who runs the game. The proof-strategy lies in making modifications to the game which $\mathcal{A}$ is unable to notice with non-negligible probability. If $\mathcal{A}$ evidently can learn nothing helpful from $\mathcal{PB}$ in the modified version, he gains the same information as if he were playing the ideal game, i.e. the game where he has no access to $\mathcal{PB}$. In this case the protocol is coercion-resistant.

The lines refer to the real coercion game as introduced in section 3.1.4. We start our modifications on line 6.

Line 6: $\mathcal{S}$ gives $\mathcal{A}$ σ instead of a random $\sigma' \in \mathbb{G}_q$. Despite the entry in IDENTIFIABLEVOTES, $\mathcal{A}$ cannot notice this difference prior to the tallying stage given $\mathrm{IND}-\mathrm{CPA}$, i.e. $\mathcal{A}$'s advantage at noticing is the same as winning the $\mathrm{IND}-\mathrm{CPA}$ game.

Line 7: Instead of $(E(\sigma), E(v_{\mathcal{A}}), \Pi_1, \Pi_2)$, $\mathcal{S}$ casts $(E(\xi_1), E(\xi_2), \Pi_1^{\mathcal{S}}, \Pi_2^{\mathcal{S}})$. The values ξ_1, ξ_2 are random from $\mathbb{G}_q$. $\mathcal{A}$'s advantage at noticing the modification prior to the tallying stage is again delimited by $\mathrm{IND}-\mathrm{CPA}$. $\mathcal{S}$ simulates two valid proofs by his control over the random oracle (refer to section 2.6). Proofs that do not hold are replaced by random proof transcripts. $\mathcal{A}$'s advantage at noticing the modification prior to the tallying stage by considering the proofs is negligibe, given HVZK.

Line 11: The same is done with the votes from voters $\mathcal{V}^{\mathcal{H}}$ as in line 7 for $\mathcal{V}^{\mathcal{A}}$'s vote.

Line 13: CASTVOTES contains nothing helpful for $\mathcal{A}$ to learn, just random values. Everything else, i.e. the votes he cast by himself, he already knows. Now we introduce modifications to the tallying stage to make sure that $\mathcal{A}$ cannot notice the previous modifications with non-negligible probability. In the end, he will learn nothing but random values from $\mathcal{PB}$. Obtaining OWNVOTES and WELLFORMEDVOTES involves no secrets and can be done in a straight-forward way. Now $\mathcal{S}$ uses d to learn the votes and the credentials posted by $\mathcal{A}$. In order to obtain UNIQUEVOTES,

he needs to perform PET over votes cast by $\mathcal{A}$ and votes cast by $\mathcal{S}$ on behalf of $\mathcal{V}^{\mathcal{H}}$ and $V^{\mathcal{A}_C}$ (lines 7 and 11). Let v_i and v_j denote two distinct votes from WellFormedVotes meant to be checked for uniqueness. If at least one out of the two is cast by $\mathcal{S}$ on behalf of $\mathcal{V}^{\mathcal{H}}$ or $V^{\mathcal{A}_C}$, $\mathcal{S}$ outputs equality in function of the original intent (considering the values of the credentials as cast prior to the first modification) and simulates a proof using the random oracle. If both $E(v_i)$ and $E(v_j)$ are cast by $\mathcal{A}$, the pair is given to the ideal PET component and the proof is published on $\mathcal{PB}$. Using the same arguments as above, $\mathcal{A}$ only has negligible advantage at noticing this modification. All he learns from UniqueVotes and the proofs is which votes cast *by him* are duplicates - something he knew from the beginning. Due to using simulated proofs and random values, there is currently no information on the other votes for him to learn. In order to obtain ValidVotes, he simulates both mixing procedures and outputs encryptions of random values including both proofs obtained using the random oracle. He simulates PET and outputs the results according to the original values of the cast votes (prior to the first modification) in random order. $\mathcal{A}$ is unable to notice these modifications on obtaining ValidVotes with non-negligible probability. Since ValidVotes contains only random values, $\mathcal{A}$ learns nothing from that set. Finally, $\mathcal{S}$ lists the plaintext-ballots according to the original values of the valid votes (prior to the modifications) in random order. He uses his control over the random oracle to obtain the proofs.

Now everything that $\mathcal{A}$ might want to learn apart from Σ and Γ is represented by random values. Therefore, his knowledge is the same as if playing the ideal coercion game.

3.3 SKHS11 Protocol

The JCJ protocol offers coercion-resistance at a high cost of effiency at tallying. As shown above, steps 3 and 4 run in $\mathcal{O}(N^2)$ time. The protocol introduced here is an attempt to reduce the running time. We propose a scheme that is δ-coercion-resistant in a paramater β, which can be chosen to yield δ as small as desired.

Some parametrizable JCJ-related protocols can be configured to achieve a degree of coercion-resistance that depends solely on the estimated Σ and Γ, just like JCJ. However, in this case, the parameters have to be chosen such that efficiency is hardly improved. In the case of these protocols, accelerating JCJ through parametrization inherently comes along with some loss in coercion-resistance in *some* respect. Nevertheless, this needs to be considered legitimate, knowing that JCJ were not coercion-resistant either if not assuming complete adversarial uncertainty regarding Σ and Γ. Most of all, it cannot be estimated, whether coercion based on these values promises less success than coercion based on the loss of coercion-resistance inherent to accelerating JCJ.

We start off in section 3.3.1 by outlining a basic proposal that - despite its simplicity - can be proved to be δ-coercion-resistant in β (we will sometimes omit δ and simply

call the schemes coercion-resistant.) One specific beneficial feature of the JCJ-scheme to be discussed further down is not incorporated to the same degree. This feature is not captured within the proposed model for assessing coercion-resistance and cannot be analysed with its help. Since we aim at preserving the security features of JCJ as much as possible, in section 3.3.2 we enhance the protocol to respect this issue anyway. The enhanced version has been published in [82]. In section 3.3.3 we provide a proof-sketch for δ-coercion-resistance of the proposal. Finally in section 3.3.4 we compare its efficiency with JCJ and provide a summary of the special features.

3.3.1 Basic Protocol

Our proposal strongly relates to the original JCJ. To reduce the quadratic running time at removing duplicates, we propose using M-PET instead of comparing all well-formed votes using the exhaustive search with PET, similarly as in [88]. This requires only seven modular exponentiations per vote (four for blinding and proof and three for decrypting and proof). For identifying the authentic votes, we suggest preserving the use of IdentifiableCredentials. Efficiency is achieved by requiring voters to indicate their entry **Cred** in the set. Thus, PET will only be applied once per vote originating from UniqueVotes. Coercion-resistance is achieved by assigning a number of noise votes (cast with a random voting credential just like $V^{\mathcal{A}_C}$'s fake vote in JCJ) to each entry in IdentifiableCredentials during the voting phase. The number X of such votes is defined by an appropriate distribution function F_X which we propose to be uniform over $[0, 2\beta]$. Each voter thus gets an average of β additional votes assigned to him.

The basic protocol is described as follows:

Registration. This is done the same way as in JCJ. Recall that voters need to know a trustworthy registrar. We can take advantage of that registrar as the source of the noise votes to be cast through the anonymous channel. In this case, voters use the untappable channel available at registration to mandate him to cast the noise votes. Alternatively, any source can be used for casting the noise votes, since no secrets are required.

Vote Casting. In addition to the tuple voters cast in JCJ, they indicate their entry **Cred** in IdentifiableCredentials. This can be done without any expensive computations, for instance by sending their identifier, **Cred** itself or its index in IdentifiableCredentials. The registrars cast noise votes assigned to the voters who placed their request at registration. They do so by using a random value **cred**$'$ from $\mathbb{G}_q$ as the voting credential. The number of noise votes is distributed according to F_X introduced above.

Tallying. At the end of the vote casting phase, CastVotes contains votes that are all connected with an entry in IdentifiableCredentials. Checking the proofs and decrypting are done as in JCJ. We start at step 3, where the unique votes are identified based on WellFormedVotes. The tallying steps are also summarized in figure 3.2.

3. Remove duplicates: The set WellFormedVotes holds elements of the shape

$(E(\hat{\sigma}_i), E(v_i))$, each of which is connected with an element from IDENTIFIABLECREDENTIALS. $\mathcal{T}$ apply M-PET on each $E(\hat{\sigma}_i)$, for all $1 \leq i \leq n_{well-formed}$. In case of two equal plaintexts, UNIQUEVOTES is obtained the same as in JCJ. Assuming voters cast one vote on average, this step runs in $\mathcal{O}((\beta+1) \cdot N)$ time.

4. **Remove Unauthentic Votes:** The tuples $(E(\hat{\sigma}_i), E(v_i), E(\sigma_i))$ are passed to a mix-net for all $1 \leq i \leq n_{unique}$. The first two elements originate from UNIQUEVOTES and the third element is the one connected from IDENTIFIABLECREDENTIALS. We call the output UNLINKABLEVOTESANDCREDENTIALS. Now $\mathcal{T}$ perform PET just on the pairs $E(\hat{\sigma}_i)$ and $E(\sigma_i)$. VALIDVOTES is constructed as in JCJ. This procedure runs in $\mathcal{O}((\beta+1) \cdot N)$ time, under the assumptions made above.

From observing CASTVOTES, $\mathcal{A}$ learns the number of cast votes related to each voter. Under the conservative assumption that there is no noise from parties other than $V^{\mathcal{A}_C}$'s trusted registrar, $\mathcal{A}$ will accept a run exactly if this number is smaller than $2\beta+1$. This probablity is $1 - \frac{1}{2\beta+1}$ in case of applying the defense strategy, and 1 when giving in to coercion. The scheme's degree of coercion-resistance is therefore determined by $\delta = \frac{1}{2\beta+1}$.

Recall that the JCJ scheme allows coercion only when given prior knowledge on Σ and Γ. Regarding attacks based on Σ, both schemes are inherently equally resistant. Regarding attacks based on Γ the basic protocol is clearly far more resistant, due to all the noise votes that are cast. However, our basic scheme allows a new attack based on the number of cast votes assigned to a voter. In this respect JCJ is more resistant against coercion attacks. Note, that attacks based on observing how many votes are cast are already captured in the JCJ model in terms of attacks based on Σ, Γ and other information on $\mathcal{PB}$. Coupling cast votes with the entries in IDENTIFIABLECREDENTIALS thus introduces an advantage for $\mathcal{A}$ specific to the basic scheme that we can easily analyse.

We now turn to the scheme's deficiency as announced above and identify a *new* attack that relates to both JCJ and the basic protocol and assess its impact on the coercion-resistance of both schemes. It is new in the sense that it affects JCJ although it cannot be analysed using the proposed model. Depending on the context, this attack may potentially be even more relevant in practice than the one inherent to the basic scheme.

Indeed, the coercion games of JCJ do not capture the temporal aspects of the vote casting phase. Yet, this may be relevant in settings where coercers and voters live together closely. More formally, we now allow $\mathcal{A}$ to observe $\mathcal{PB}$ throughout the protocol and retain $V^{\mathcal{A}_C}$'s access to the anonymous channel during a given period of time. This gives $\mathcal{A}$ an advantage at finding out whether $V^{\mathcal{A}_C}$ cast his vote during a (possibly rare) moment of privacy. Let t denote the timespan given for casting votes and Δt the time given to $V^{\mathcal{A}_C}$ to access the anonymous channel. Further, let the number of votes $X_{\Delta t}$ cast by $N_{\mathcal{H}}$ honest voters during Δt be uniformly distributed over $[0, 2 \cdot \frac{\Delta t}{t} \cdot N_{\mathcal{H}}]$, thus assuming that voters cast one vote on average during t. For simplicity, we assume $2 \cdot \frac{\Delta t}{t} \cdot N_{\mathcal{H}}$ to be an integer. Then $\mathcal{A}$ will accept a run only if the actual number of votes cast during that time $x_{\Delta t}$ satisfies $x_{\Delta t} \leq 2 \cdot \frac{\Delta t}{t} \cdot N_{\mathcal{H}}$. The corresponding degree of

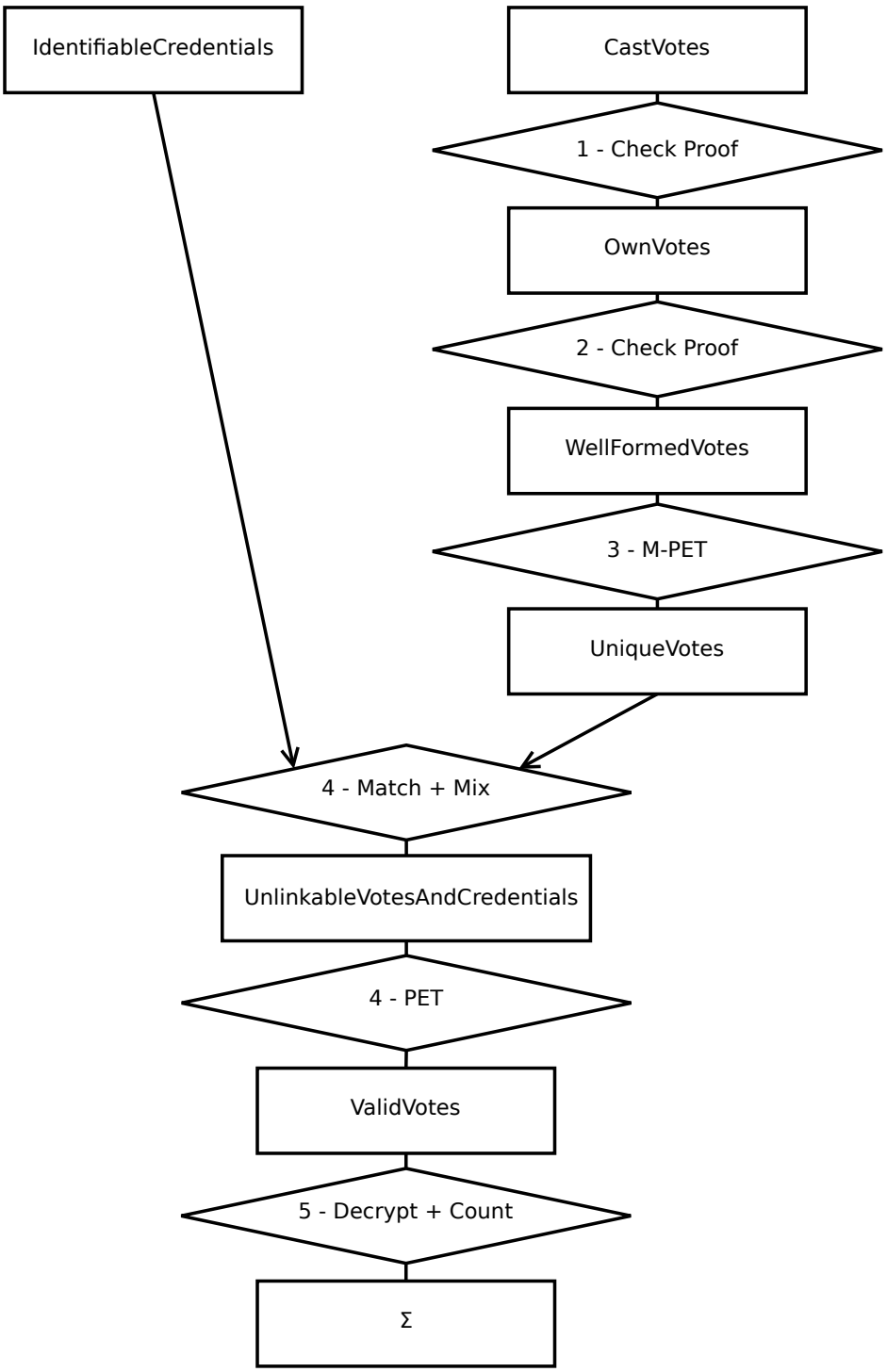


Figure 3.2: Tallying phase of the SKHS11 basic protocol.

coercion-resistance $\delta_{\Delta t,\mathcal{JCJ}}$ based on the attack is $\frac{1}{2\cdot\frac{\Delta t}{t}\cdot N_{\mathcal{H}}+1}$.

In the basic protocol, $\mathcal{A}$ does not observe $\mathcal{PB}$ to find $V^{\mathcal{A}_C}$'s vote among the the votes

cast by honest voters, but rather among the number of votes cast for $V^{\mathcal{A}_C}$ by his trusted registrar. This number is uniformly distributed over $[0, \frac{\Delta t}{t} \cdot 2\beta]$, whereas β needs to be chosen smaller than $N_{\mathcal{H}}$ in order to be more efficient than JCJ. Following the same arguments as above, we conclude that the degree of coercion-resistance $\delta_{\Delta t,basic}$ related to the attack is $\frac{1}{2 \cdot \frac{\Delta t}{t} \cdot \beta + 1}$ in the basic protocol. We observe the relation $\frac{\delta_{\Delta t,\mathcal{JCJ}}}{\delta_{\Delta t,basic}}$. Intuitively speaking, it says by which factor JCJ is more coercion-resistant regarding the observed *temporal attack*. If we set $\frac{\Delta t}{t} \cdot N_{\mathcal{H}} \geq 1$, which seems sensible for large settings, apparently we have $\frac{\beta}{N_{\mathcal{H}}} \leq \frac{\delta_{\Delta t,\mathcal{JCJ}}}{\delta_{\Delta t,basic}} \leq \frac{2}{3} \cdot \frac{\beta}{N_{\mathcal{H}}} + \frac{1}{3}$.

In the next section we will make a slight modification on the basic protocol to obtain the enhanced version. It performs much better regarding the temporal attack than JCJ.

3.3.2 Enhanced Protocol

We aim at improving the basic protocol to be more resistant against temporal attacks. We do so by making a slight enhancement that hardly comes along with any perceivable efficiency drawbacks. The enhacement renders our proposal even more coercion-resistant than JCJ with temporal attacks. Particularly, $V^{\mathcal{A}_C}$'s vote will be hidden not just among his own noise votes, but additionally among all other noise votes cast to $\mathcal{PB}$ too.

By having voters encrypt the indication of their public credential **Cred** in IDENTIFIABLECREDENTIALS, $\mathcal{A}$ cannot perform his attack during the phase of vote casting. By making sure that the indications are only decrypted once being unlinkable with the (presumed) timestamps in CASTVOTES, $\mathcal{A}$ will never be able to link votes to the timespan Δt of $V^{\mathcal{A}_C}$'s granted access to the anonymous channel. In this case, $\mathcal{A}$'s strategy reduces to the one in JCJ, i.e. where he observes the number of votes cast during Δt without any auxilliary information apart from their distribution.

Registration is done the same way as in the basic protocol. The first modification affects vote casting.

Vote Casting. This time V enrypts the indicator to his entry **Cred** in IDENTIFIABLECREDENTIALS. We take this indication to be the index # of **Cred** within the set. For simplicity, we pretend that $\# \in \mathbb{G}_q$. The information cast to CASTVOTES now takes the shape $(E(\sigma), E(v), E(\#), \Pi_1, \Pi_2, \Pi_3)$. Additionally to σ and v, V also needs to prove knowledge of #. He does so by casting the proof Π_3. The new proof is needed for the same reasons as the other two. This is explained within the description of the JCJ protocol in section 3.2.1.

Tallying. The three proofs are checked straightforwardly to obtain the set WELLFORMEDVOTES. From there, duplicates are ruled out using M-PET, just like in the basic protocol. The tallying steps are also summarized in figure 3.3.

3. **Remove duplicates**: The set WELLFORMEDVOTES holds elements of the shape $(E(\hat{\sigma}_i), E(v_i), E(\#_i))$. For all $1 \leq i \leq n_{well-formed}$, $\mathcal{T}$ apply M-PET on $E(\hat{\sigma}_i)$ and generate UNIQUEVOTES just like in the basic protocol. So far, the adversary was not given any non-negligible advantage from posting $E(\#_i)$ and Π_3 as compared with JCJ. The new value $E(\#_i)$ has been ignored. This is clearly a necessary

condition for improving on the temporal attacks. Before decrypting the indices, we need to make sure that the adversary cannot relate them back to their time of being cast.

4. **Remove Unauthentic Votes:** $\mathcal{T}$ pass UNIQUEVOTES to a mix-net. They hereby detach the votes from the information on when they were cast. We call the output DEHISTORIZEDVOTES, whose elements take the shape $(E(\hat{\sigma}_i), E(v_i), E(\#_i))$. Now all values $E(\#_i)$ are decrypted. For all i where $\#_i = j$ for any $j \in \{1, ..., N\}$, $\mathcal{T}$ form the tuple $(E(\hat{\sigma}_i), E(v_i), E(\sigma_j))$, where the third element is the one connected from IDENTIFIABLECREDENTIALS. Now $\mathcal{T}$ carry on with this tallying step like in the basic protocol.

In the enhanced setting, the number of votes cast by honest voters and all trusted registrars is uniformly distributed over $[0, 2 \cdot \frac{\Delta t}{t} \cdot (N_{\mathcal{H}} + N_{\mathcal{H}} \cdot \beta)]$ when following the argumentation above. This leaves us with $\delta_{\Delta t, enhanced} = \frac{1}{2 \cdot \frac{\Delta t}{t} \cdot (N_{\mathcal{H}} + N_{\mathcal{H}} \cdot \beta) + 1}$ for the enhanced protocol regarding the temporal attack. As above, we assume $\frac{\Delta t}{t} \cdot N_{\mathcal{H}} \geq 1$ and find that indeed $\frac{1}{1+\beta} \leq \frac{\delta_{\Delta t, enhanced}}{\delta_{\Delta t, JCJ}} \leq \frac{1}{1+\frac{2}{3} \cdot \beta}$. This shows that the enhanced protocol is this clearly more resistant against temporal attacks than JCJ.

So far we have shown that the enhanced protocol performs better than JCJ with regard to temporal attacks. We have noted that attacks based on prior knowledge on Σ are equally promising in both schemes, whereas the enhanced protocol performs much better regarding attacks based on Γ. By setting $\beta = \frac{\Delta t}{t} \cdot N_{\mathcal{H}}$ it is easy to show that an attack based on the number of votes assigned to a voter in the enhanced scheme is equally promising as a temporal attack in JCJ. These observations make us wonder whether the enhanced protocol could even be shown to be as coercion-resistant as JCJ in a reasonable model. For future work, we propose to observe attack strategies that use a combination of the ones discussed here. Clearly, correlations do exist. The schemes should be compared again based on the results. For now, we return to the model for assessing coercion-resistance introduced in section 3.1.4 and provide a proof sketch for the enhanced scheme.

3.3.3 Proof Sketch for Coercion-Resistance

The argumentation in the proof is very similar as in JCJ. Again, we start from the game run by simulator $\mathcal{S}$ according to the protocol. By making modifications to that game we end up in a game where all information that is interesting for $\mathcal{A}$ is represented by encryptions of random values. For simplicity, we will simply outline the differences that emerge from aspects in which the schemes differ. We thus start at line 7. Also for simplicity, we assume that the noise votes cast by the trusted registrar are cast along with the votes from the honest voters $\mathcal{V}^{\mathcal{H}}$ on line 11.

Line 7: Instead of $(E(\sigma), E(v_{\mathcal{A}}), E(\#), \Pi_1, \Pi_2, \Pi_3)$, $\mathcal{S}$ casts $(E(\xi_1), E(\xi_2), E(\xi_3), \Pi_1^{\mathcal{S}}, \Pi_2^{\mathcal{S}}, \Pi_3^{\mathcal{S}})$. Thus the same arguments apply at modifying the new values $E(\#)$ and Π_3 as for the other four values in the proof of the JCJ protocol.

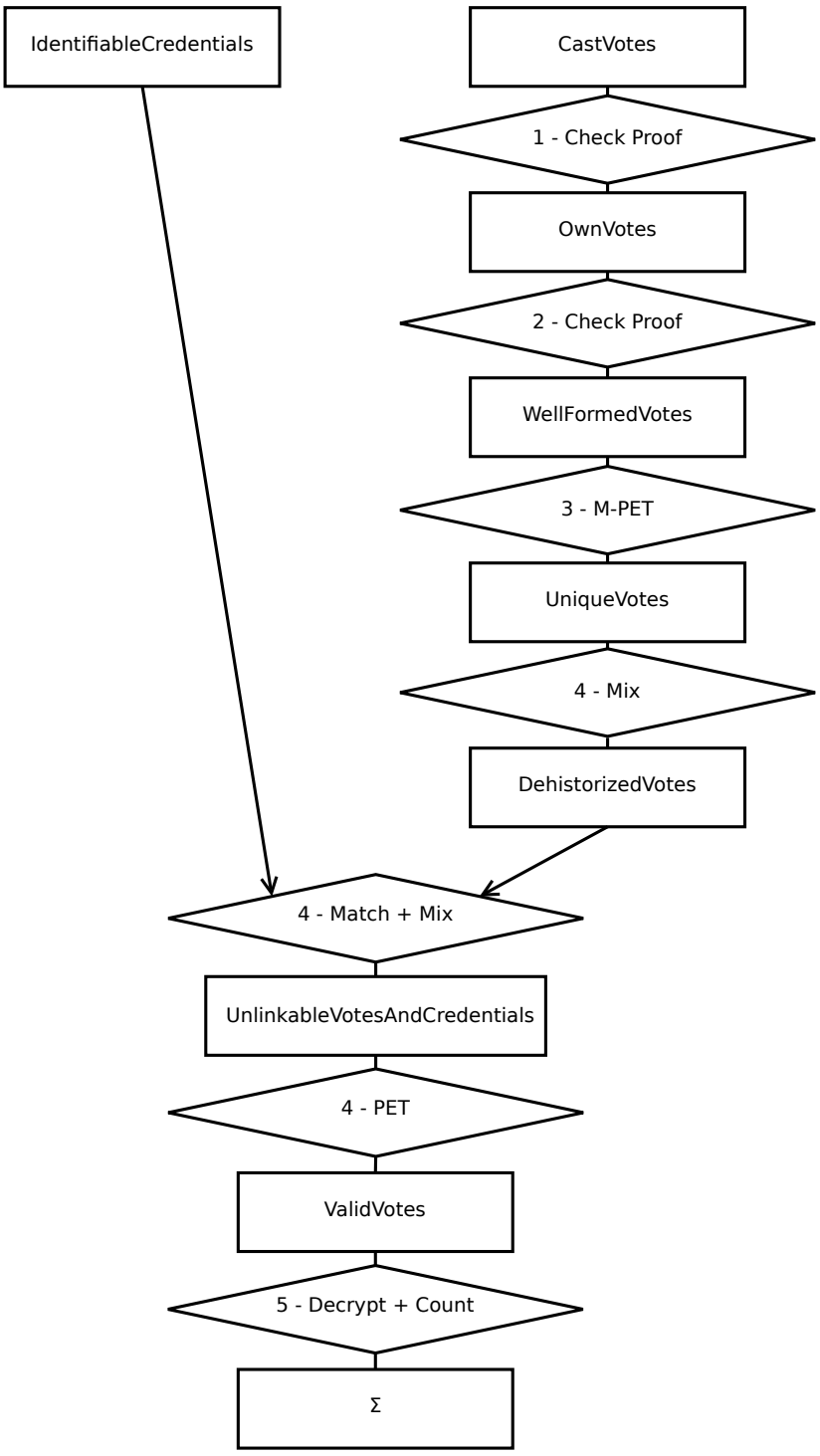


Figure 3.3: Tallying phase of the SKHS11 protocol.

Line 11: The modification now also applies for votes cast by $\mathcal{V}^{\mathcal{A}}$'s trusted registrar. Their number is X, which is a uniformly distributed random variable as described

above. Now in case $b = 0$ and $X = 2 \cdot \beta$, $\mathcal{V}^{\mathcal{H}}$'s trusted registrar casts only $2 \cdot \beta - 1$ votes. $\mathcal{A}$ notices this modification with a probability of δ prior to the tallying phase.

Line 13: We start from the set WELLFORMEDVOTES that still contains duplicates. It holds elements of the shape $(E(\hat{\sigma}_i), E(v_i), E(\#_i))$ for votes cast by $\mathcal{A}$ and $(E(\xi_{1,i}, E(\xi_{2,i}), E(\xi_{3,i})$ for votes cast on behalf of the honest voters $\mathcal{V}^{\mathcal{H}}$ and from $V^{\mathcal{A}_C}$. $\mathcal{S}$ knows all encrypted values and in case of the random plaintexts also the original values prior to the first modification, i.e. the plaintexts encrypted by the honest voters. Now, he obtains UNIQUEVOTES. For the elements originally cast by $\mathcal{A}$, $\mathcal{S}$ performs M-PET by following the behaviour of the actual component. Thus, $\mathcal{A}$ cannot learn anything additional, since he encrypted the values himself. For the other votes $\mathcal{S}$ simulates M-PET according to the original intent of the honest voters. He simulates the proof using his control over the random oracle. $\mathcal{S}$ obtains the set DEHISTORIZEDVOTES just like UNLINKABLEVOTES in JCJ. All elements from DEHISTORIZEDVOTES now all comprise new random elements and take the shape $(E(\xi'_{1,i}), E(\xi'_{2,i}), E(\xi'_{3,i}))$. $\mathcal{S}$ simulates decrypting the elements $E(\xi'_{3,i})$ and outputs the values $\#_i$ according to the original intent, just as when decrypting the votes in the JCJ proof above. Now the same arguments apply when obtaining VALIDVOTES and the final tally Σ.

3.3.4 Efficiency and Other Properties

As shown in the text above, the proposal manages to reduce the running time significantly at tallying, particularly for large N. The gain at efficiency and the degree of coercion-resistance in the sense of the proposed model can be controlled and expressed in terms of β. The following table is meant to give a sense of the efficiency features of the new scheme in comparison with JCJ. It shows the number of modular exponentiations in function of N and δ, where $N = 1000$ and $N = 100'000$. We choose $\beta = 49.5$ and $\beta = 9.5$, thus yielding degrees of coercion-resistance of $\delta = \frac{1}{100}$ and $\delta = \frac{1}{20}$ respectively for the new scheme. For simplicity we assume the participation of only one trustee at each distributed operation and a set $\mathcal{C}$ of three voting options (such as *yes*, *no* and *empty*). For proofs generated by the trustees, we only consider their generation, not the verification. We assume that all voters cast exactly one valid vote. Instead of the precise numbers we show the next higher integer to the logarithm in base 2. For the generic terms refer to [74].[7] The table shows that the payoff in terms of efficiency is tremendous, particularly for large N, where tallying is accelerated roughly by the factor of 1000.

[7] We differ in some of the proposed terms: 1. The paper proposes to require $2 \cdot N$ modular exponentiations at creating proofs of correct re-encryption at RPC. We assume that this is done by revealing the employed randomness. In this case there are no computational costs on the server side worth mentioning. 2. We believe that eliminating invalid votes in SKHS11 takes $4N(m+1)(\beta+1)$, where m denotes the number of voting options. (Here we use our notation and assume that the number of votes equals the number of voters.) 3. Also it seems that inserting noise votes takes $\beta N(4m+6)$ instead of $6\beta N$ modular exponentiations.

Scheme	JCJ	SKHS11	SKHS11	JCJ	SKHS11	SKHS11
N - Number of voters	1000	1000	1000	100′000	100′000	100′000
δ - Degree of c-resistance	0	$1/100$	$1/20$	0	$1/100$	$1/20$
Vote casting	4	5	5	4	5	5
Tallying - check proofs	14	20	18	21	27	25
Tallying - rem. duplicates	22	19	17	36	26	23
Tallying - rem. unauth.	23	21	19	37	28	26

We refer to the previous section where we argue that the new scheme is likely to have a similar degree of coercion-resistance as JCJ, when allowing temporal attacks and when not assuming complete adversarial uncertainty regarding Σ and Γ. However, these assumptions are not captured by the model proposed by JCJ for assessing coercion-resistance.

There are more beneficial features that are shared by the two schemes. First of all, they share equal degrees of verifiability. The assessment for JCJ from section 3.2.2 also applies for the new proposal. Second, in both schemes it is possible for voters to re-use their credentials without any intermediate communication with the trustees. Third, both schemes respect coercion-resistance at credential retention. When voters move away, their credential **Cred** can simply be marked as *invalid* in IdentifiableCredentials and left ignored at tallying. The adversary will not notice whether he has previously received the correct voting credential. Finally, voters who want to abstain in the first place also enjoy full protection from coercers and vote-buyers as in JCJ. Since they do not cast their vote, the adversary will never be able to tell whether they applied their defense strategy. This can easily be observed by ignoring the instruction on line 7 of the real coercion game and waiving the corresponding modifications in the proof.

3.4 SKHS12 Protocol

The previous protocol improves the JCJ protocol's performance at tallying significantly. Although implementations that offer a sufficient degree of coercion-resistance now appear to be feasible on a large scale, the tallying phase may still seem to take long for big values of N. The protocol introduced in the present section aims at reducing the computation time at tallying even more. As in the previous protocol, there is a parameter β that underlies the degree of coercion-resistance δ. Again, large values of β imply small values of δ at the cost of computational time. Although this protocol too has phases that require a lot of computing, at least noone is kept waiting at any stage.

We start off again by introducing a basic protocol in section 3.4.1. It is δ-coercion-resistant and verifiable in the sense of the definition in [49], however, it lacks *eligibility verifiability* (refer to section 3.2.2 for details). In order to provide eligibility verifiability just as JCJ does, we introduce an enhanced version of the protocol in section 3.4.2. Both versions of the protocol have been published in [83]. In section 3.4.3 we provide a proof-sketch for δ-coercion-resistance of the proposal. Finally, in section 3.4.4 we compare its efficiency with JCJ and SKHS11 and provide a summary on the special features.

3.4.1 Basic Protocol

In the SKHS11 protocol the voters assign their votes to their credential in IDENTIFIABLECREDENTIALS on $\mathcal{PB}$. The credential from IDENTIFIABLECREDENTIALS and the information cast by the voters are eventually matched and jointly processed for authentication. However, we need to rely on noise votes that keep the adversary in the dark as to whether the voter under coercion has cast his vote or not. Dealing with these noise votes may still render tallying slower than desired. In the new protocol, the voters assign their votes to their credential **Cred** on $\mathcal{PB}$ too. However this time, the credential is not in IDENTIFIABLECREDENTIALS, but in UNLINKABLECREDENTIALS, i.e. it can not be linked to the voter's identity. The adversary's best coercion-strategy lies in observing whether there have been two votes assigned to the same credential that the voter under coercion claims to be his. Since UNLINKABLECREDENTIALS holds at least $\beta \cdot N$ credentials, his success probability is low - specifically $\delta < \frac{1}{\beta}$, as will be shown further down. Tallying is done the same way as in the basic version of SKHS11. However, creating UNLINKABLECREDENTIALS (at *post-registration*) and giving the voters the ability to lie about which one their entry in UNLINKABLECREDENTIALS is (at *pre-registration*) both require some extra efforts prior to voting.

The basic protocol is described as follows:

Pre-Registration. The registrars $\mathcal{R}$ initialize the list IDENTIFIABLECREDENTIALS to hold a number of $\beta \cdot N_+$ credentials **Cred**. β is chosen according to the desired tradeoff between coercion-resistance and efficiency and N_+ is the maximum number of individual voters ever to participate at votes run by the system. Since credentials may not be reused by different voters, N_+ should be chosen high enough to anticipate new community members that register after the first votes have taken place. For each of the potential N_+ voters, $\mathcal{R}$ jointly compute the random value $\sigma \in \mathbb{G}_q$ and keep their shares to themselves, just as in JCJ. Additionally, they proceed the same way to obtain another random value $\# \in \mathbb{G}_q$ for each potential voter. V's voting credential **cred** will be a tuple $(\sigma, \#)$. Its public counterpart $(E(\sigma), E(\#))$ is an element **Cred** of IDENTIFIABLECREDENTIALS appended during pre-registration. After voters register, they need to be able to lie to the adversary $\mathcal{A}$ not only about σ but also about $\#$. Since during the tallying stage all $\#$-values will need to be public on $\mathcal{PB}$, the voters need to be able to select an existing $\#$ value as their fake $\#'$ at registration already. To this end, all $\#$-components are passed to a mix-net and decrypted to form the list UNLINKABLE#CREDENTIALS.[8] This step only needs to be performed prior to the first voting event hosted by the system, particularly before the first voter registers.

Registration. The registrars choose an unassigned credential **cred** that has been prepared during pre-registration and hand both components to the voter the same way as in JCJ. They associate **Cred** with an identifier of V, such as name, birthday and address in IDENTIFIABLECREDENTIALS. In order to lie to the coercer about the value of $\#$, the voter needs to be able to give him a value $\#'$ that is a value actually used as

[8] [83] proposes to have both components of each entry **Cred** from IDENTIFIABLECREDENTIALS processed pair-wise in a mix-net. Although this does not hurt, it is yet unnecessary.

the #-component of an arbitrary entry in IDENTIFIABLECREDENTIALS. To this end, he randomly selects and memorizes a value from the list UNLINKABLE#CREDENTIALS.

Post-Registration. In this step the list UNLINKABLECREDENTIALS is generated. It serves the same purpose at tallying in the present protocol as IDENTIFIABLECREDENTIALS in SKHS11. Particularly, the entries **Cred** from list IDENTIFIABLECREDENTIALS are processed pair-wise by a mix-net. Afterwards the #-components are decrypted. The post-registration step needs to be completed only prior to tallying, i.e. the phase in which voters cast their votes can be used for this step. Thereby, the negative impact of the time-consuming mix-nets is mitigated, or even fully compensated, given that the voting phase is sufficiently long. [9] [10]

Vote Casting. In addition to the tuple voters cast in JCJ, they give an indication of their entry **Cred** in UNLINKABLECREDENTIALS. They thus form the tuple $(E(\sigma), E(v), \#, \Pi_1, \Pi_2)$ and append it to CASTVOTES on $\mathcal{PB}$.

Tallying. This step is conducted the same as in the basic protocol of SKHS11, just that we use the list UNLINKABLECREDENTIALS instead of IDENTIFIABLECREDENTIALS. At the end of the vote casting phase, CASTVOTES contains votes that are all connected with an entry in UNLINKABLECREDENTIALS. Checking the proofs is done as in JCJ. We start at step 3, where the unique votes are identified based on WELLFORMEDVOTES. The tallying steps are also summarized in figure 3.4.

3. **Remove duplicates**: The set WELLFORMEDVOTES holds elements of the shape $(E(\hat{\sigma}_i), E(v_i))$, each of which is connected with and element from UNLINKABLECREDENTIALS. $\mathcal{T}$ apply M-PET on each $E(\hat{\sigma}_i)$, for all $1 \leq i \leq n_{well-formed}$. In case of two equal plaintexts, UNIQUEVOTES is obtained the same as in JCJ. Assuming voters cast one vote on average, this step runs in $\mathcal{O}(N)$ time, as opposed

[9] Since UNLINKABLECREDENTIALS contains the #-components in plaintext, post-registration can be performed prior to registration instead of pre-registration at the first voting event. This option has not been pointed out in [83]. However, if at subsequent voting events the registration phase is used for retaining the credentials of voters who are no longer eligible, post-registration may only start after the registration phase. Yet, it only needs to finish prior to tallying. How to perform the retention of the credentials of leaving voters without compromising coercion-resistance is discussed in section 3.4.4.

[10] In [83] it is not discussed whether post-registration needs to be performed at every voting event or just within the first one, when assuming that the electorate remains unchanged, i.e. assuming that no voters move away or join the community (thus $N_+ = N$). The way post-registration is defined, it would not make sense to repeat the step at every voting event, since the #-components remain unchanged. However, one may consider the risk that after a few voting events there may appear some correlations between the #-components to which votes are assigned on one hand and the final tally on the other. Depending on how this risk regarding the requirement *fairness* is assessed, the protocol could easily be enhanced to provide different values for the #-components at each voting event without requiring the untappable channel from the registration step again. A solution could relate to the technique shown in [80]. However, we note that there are many renowned verifiable voting schemes that allow the public to witness even which particular voters have participated at a voting event. These allow even better predicitions of the final tally. Yet, in any case the issue needs to be addressed when introducing verifiable Internet voting in practice. Clearly, if the electorate *does* change between voting events, post-registration needs to be performed each time.

to $\mathcal{O}((\beta+1)N)$ in SKHS11 and $\mathcal{O}(N^2)$ in JCJ.

4. **Remove Unauthentic Votes:** The tuples $(E(\hat{\sigma}_i), E(v_i), E(\sigma_i))$ are passed to a mix-net for all $1 \leq i \leq n_{unique}$. The first two elements originate from UNIQUEVOTES and the third element is the one connected from UNLINKABLECREDENTIALS. We call the output UNLINKABLEVOTESANDCREDENTIALS. Now $\mathcal{T}$ perform PET just on the pairs $E(\hat{\sigma}_i)$ and $E(\sigma_i)$. VALIDVOTES is constructed as in JCJ. This step too runs in $\mathcal{O}(N)$ time, as opposed to $\mathcal{O}((\beta+1)N)$ in SKHS11 and $\mathcal{O}(N^2)$ in JCJ.

In his defense strategy, voter $V^{\mathcal{A}_C}$ lies to $\mathcal{A}$ about both components of the credential $(\sigma, \#)$ and hands him $(\sigma', \#')$ instead, where σ' is chosen at random from $\mathbb{G}_q$ and $\#'$ is random from UNLINKABLE#COMPONENTS, where $\#' \neq \#$. If he is fortunate enough to choose $\#'$ such that it is not the #-component of an other voter in $\mathcal{V}$, then $\mathcal{A}$ can not learn from CASTVOTES whether $V^{\mathcal{A}_C}$ has cast his vote or not: $(\sigma', \#')$ is a credential that is not used by anybody and could just as well be the one from $V^{\mathcal{A}_C}$. Further, when following the JCJ model from section 3.1.4, due to adversarial uncertainty regarding Σ and Γ, $\mathcal{A}$ will not notice whether or not $V^{\mathcal{A}_C}$ has cast his vote using the actual credential $(\sigma, \#)$. On the other hand, if $V^{\mathcal{A}_C}$ is unlucky and $\#'$ is the #-component of an other voter's credential, $\mathcal{A}$ may find that two votes have been cast using $\#'$ and thus conlude that $V^{\mathcal{A}_C}$ has applied his defense strategy. The probability of choosing such an unfortunate value as $\#'$ prior to the first voting event is $\delta = \frac{N_+ - 1}{\beta \cdot N_+ - 1}$ and thus, more simply put, $\delta < \frac{1}{\beta}$.

We now turn to the deficiency regarding verifiability in the basic scheme and provide an enhancement to overcome the issue.

3.4.2 Enhanced Protocol

The basic protocol fulfills individual verifiability, i.e. voters are able to verify that their vote has been cast as intended, recorded as cast and tallied as recorded. Also it fulfills universal verifiability, in the sense that the public can detect the exclusion of legitimate votes, changes to legitimate votes and the inclusion of multiple votes cast with the same credential. Regarding verifiability, our basic scheme is thus not less powerful than the coercion-resistant scheme by Araújo et al. [3, 5].[11] However, as mentioned in the JCJ paper, it may be desirable for any election observer to verify that credentials have only been assigned to voters whose names are on a published roll. We have defined *universal verifiability* to accomodate this notion. Indeed, the JCJ-protocol provides this kind of verifiability, which is in effect *eligbility verifiability* or as put in [83], *improved verifiability*. For our basic protocol to respect eligibility verifiability, we need to assume trustworthy majorities among registrars and talliers. In order to detect the event of colluding registrars or talliers that cast votes with an unassigned credential enlisted in IDENTIFIABLECREDENTIALS, we propose a simple enhancement to step 4 of tallying. All tallying steps are also summarized in figure 3.5.

Tallying.

[11] The same scheme has recently been improved in [7] to respect eligibility verifiability.

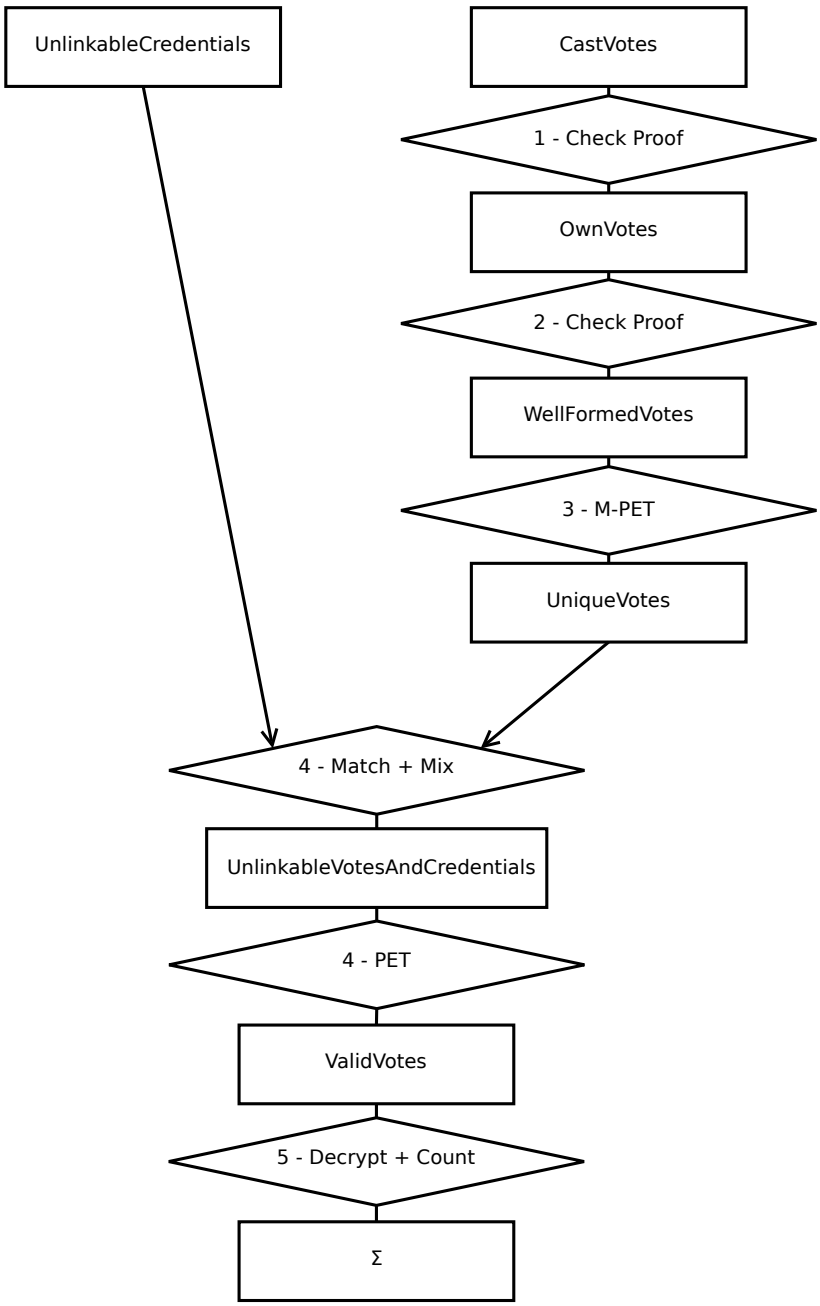


Figure 3.4: Tallying phase of the SKHS12 basic protocol.

4. **Remove Unauthentic Votes:** The tuples $(E(\hat{\sigma}_i), E(v_i), E(\sigma_i))$ are passed to a mix-net for all $1 \leq i \leq n_{unique}$. The first two elements originate from UNIQUEVOTES and the third element is the one connected from UNLINKABLECREDENTIALS. We call the output UNLINKABLEVOTESANDCREDENTIALS. Now $\mathcal{T}$ perform PET just on the pairs $E(\hat{\sigma}_i)$ and $E(\sigma_i)$ of that list. The first change to the tallying step affects how votes are dealt with for which PET reveals $\hat{\sigma}_i = \sigma_i$. Instead of appending

$E(v_i)$ to ValidVotes, we append $(E(\hat{\sigma}_i), E(v_i))$ to a new list that we call UnalteredVotes. This list may still contain votes cast with a credential enlisted in IdentifiableCredentials (and UnlinkableCredentials) that has not been assigned to any voter. In order to rule out these votes from UnalteredVotes, we consider the sublist of IdentifiableCredentials which holds the credentials that *have been* assigned to the voters. We call this list IdentifiableAssignedCredentials. The σ-components of this list are processed by a mix-net to obtain the list UnlinkableAssignedσComponents. Now the talliers perform M-PET on the elements $E(\sigma_i)$ of UnlinkableAssignedσComponents and the $E(\hat{\sigma}_j)$-components of the elements in UnalteredVotes. If M-PET reveals $\sigma_i = \hat{\sigma}_j$ for any i and j, then $E(v_j)$ is appended to ValidVotes. Note that the conditions to perform M-PET are given at this stage, due to logarithms of the encrypted inputs being unknown (refer to section 3.1.1).

3.4.3 Proof Sketch for Coercion-Resistance

Again, the argumentation is very simililar as with the previous protocols. We have the simulator $\mathcal{S}$ run the real coercion game according to the protocol. By making modifications to that game we end up in a game where all information that is interesting for $\mathcal{A}$ is represented by encryptions of random values. For simplicity, we will just outline the differences that emerge from aspects in which the present scheme differs from JCJ. We thus start at line 6. We relate our exposition to the techniques applied in the proof sketch for SKHS11 in order to avoid repetitions. Also for simplicity, we allow $\mathcal{S}$ to run pre-registration and post-registration along with the registration step which is instructed on line 2 of the real coercion game.

Line 6: $\mathcal{S}$ gives $\mathcal{A}$ his real credential $(\sigma, \#)$ instead of a $(\sigma', \#')$. Despite the entry in IdentifiableVotes, $\mathcal{A}$ cannot notice the difference in the σ-components given $\mathrm{IND}-\mathrm{CPA}$ prior to tallying, i.e. $\mathcal{A}$'s advantage at noticing is the same as winning the $\mathrm{IND}-\mathrm{CPA}$ game. However, he would notice the difference based on the #-components prior to tallying. This issue is solved by the modification on line 7.

Line 7: Instead of $(E(\sigma), E(v_{\mathcal{A}}), \#, \Pi_1, \Pi_2)$, $\mathcal{S}$ casts $(E(\xi_1), E(\xi_2), \xi_3, \Pi_1^{\mathcal{S}}, \Pi_2^{\mathcal{S}})$. He chooses the values $E(\xi_1)$, $E(\xi_2)$, $\Pi_1^{\mathcal{S}}$ and $\Pi_2^{\mathcal{S}}$ just like in JCJ. ξ_3 is set to a #-component of an element in IdentifiableCredentials, which has not been assigned to any voter. Thus, $\mathcal{A}$ cannot notice the modification in line 6 prior to tallying. However, if $V^{\mathcal{A}_C}$ is unfortunate enough to choose a value # that is used by a different voter, then $\mathcal{A}$ will notice the modification. The probability of such an event is δ.

Line 13: Again, CastVotes contains nothing helpful for $\mathcal{A}$ to learn, just random values. Everything else, i.e. the votes he cast by himself, he already knows. Now we introduce modifications to the tallying stage to make sure that $\mathcal{A}$ cannot notice the previous modifications with non-negligible probability. In the end, he will

Figure 3.5: Tallying phase of the SKHS12 protocol.

learn nothing but random values from $\mathcal{PB}$. In order to obtain OwnVotes, WellFormedVotes, UniqueVotes, UnalteredVotes and UnlinkableAssignedσComponents, $\mathcal{S}$ applies the same strategies as in the modifications of the previous protocols. In order to obtain ValidVotes, $\mathcal{S}$ simulates M-PET on the input of all $E(\sigma_i)$ from UnlinkableAssignedσComponents and all $E(\hat{\sigma}_j)$-components from Un-

ALTEREDVOTES. He simulates the output of M-PET according to the values from CASTVOTES prior to the first modification. $\mathcal{A}$ cannot learn anything from this modification, since $\sigma_i = \hat{\sigma}_j$ for some i and j are the only cases where a discrete logarithm of a plaintext is known in the base of another. For obtaining the final tally Σ, $\mathcal{S}$ follows the same strategy as in JCJ.

3.4.4 Efficiency and Other Properties

After SKHS11, the SKHS12 protocol allows another significant reduction of the running time at tallying. The computational cost is payed during the steps of pre-registration and post-registration. This generally seems less critical, since both steps can be conducted prior to the phase where voters cast their votes. Optionally, the time of the vote casting phase can be used to finish post-registration, since its output is only needed at tallying. By starting both pre-registration and post-registration early enough, noone will be kept waiting at any stage.

In the following tables, we compare the performance of SKHS12 with JCJ and SKHS11. They express the number of modular exponentiations (the next higher integer to the logarithm in base 2) in function of N and δ. We use the same premises as in section 3.3.4. For the case where $N = 1000$, we set $N_+ = 2000$ and accordingly for $N = 100'000$ we set $N_+ = 200'000$. Note that given $\beta = \frac{N_+ + \delta - 1}{N_+ \cdot \delta}$, the value $\beta \cdot N_+$, which is the number of entries in INDENTIFIABLECREDENTIALS, is an integer when considering $\delta = \frac{1}{20}$ and $\delta = \frac{1}{100}$. Specifically, if $N = 1000$ the size of INDENTIFIABLECREDENTIALS is $39'981$ for $\delta = \frac{1}{20}$ and $199'901$ for $\delta = \frac{1}{100}$. If $N = 100'000$ the size is $3'999'981$ for $\delta = \frac{1}{20}$ and $19'999'901$ for $\delta = \frac{1}{100}$.

Scheme	JCJ	SKHS11	SKHS12	SKHS11	SKHS12
N - Number of voters	1000	1000	1000	1000	1000
δ - Degree of c-resistance	0	$1/100$	$1/100$	$1/20$	$1/20$
Pre-Registration	-	-	22	-	19
Post-Registration	-	-	22	-	19
Vote casting	4	5	4	5	4
Tallying - check proofs	14	20	14	18	14
Tallying - rem. duplicates	22	19	13	17	13
Tallying - rem. unauth.	23	21	16	19	16

As implied above, unlike SKHS11, the tallying phase of SKHS12 is completely insensitive to the desired degree of coercion-resistance δ, which is particularly interesting if a small value for δ is desired. For both schemes, tallying scales linearly over the number of participating voters N. These notions can be observed in the tables. Also, we can verify that the cost of pre-registration and post-registration of SKHS12 are approximately as time-consuming as tallying in SKHS11.

Scheme	JCJ	SKHS11	SKHS12	SKHS11	SKHS12
N - Number of voters	100′000	100′000	100′000	100′000	100′000
δ - Degree of c-resistance	0	$1/100$	$1/100$	$1/20$	$1/20$
Pre-Registration	-	-	28	-	26
Post-Registration	-	-	28	-	26
Vote casting	4	5	4	5	4
Tallying - check proofs	21	27	21	25	21
Tallying - rem. duplicates	36	26	20	23	20
Tallying - rem. unauth.	37	28	22	26	22

We now turn to the special properties of the protocol. Recall from section 3.3 that the SKHS11 protocol is particularly resistant against temporal attacks. However, in SKHS12 the value # is sent in plaintext. Clearly, this leaves the adversary $\mathcal{A}$ room for temporal attacks, i.e. attacks that that cannot be captured by the JCJ-model introduced in section 3.1.4 for assessing coercion-resistance. Particularly, $\mathcal{A}$ could retain the anonymous channel from voter $V^{\mathcal{A}_C}$ for most of the time and observe $\mathcal{PB}$ during the remaining time of the vote casting phase. Clearly, at the first voting event $V^{\mathcal{A}_C}$ is just as safe as when using JCJ. However in each subsequent vote, $\mathcal{A}$ can filter the values # on UnlinkableCredentials according to the votes cast during the timespan of $V^{\mathcal{A}_C}$'s access to the anonymous channel. He thus has a strategy to figure out whether $\#' = \#$ and, even more, to figure out the value of the true #-component of the voter's credential. In order to address temporal attacks as in SKHS11, the present protocol could easily be modified to have voters cast an encryption of # and a corresponding proof Π_3 of knowing #. At tallying, the cast votes could be de-historized before matching the votes with the credentials from UnlinkableCredentials as in SKHS11. This would not have a significant impact on the scheme's efficiency. (The value 22 for removing unauthentic votes at tallying would change to 23 in the case of $N = 100'000$. This is the only value in the tables that would change due to this enhancement.)

As in JCJ, voters should be able to re-use their crentials at subsequent voting events without any intermediate communication with the trustees. We therefore need to explore how credential retention can be done in SKHS12 without compromising coercion-resistance. Particularly, even after moving away, $V^{\mathcal{A}_C}$ should not need to fear that $\mathcal{A}$ will be able to tell whether he has previously received the correct credential **cred** or a fake credential **cred**′. In JCJ and SKHS11, retaining credentials by ignoring the respective entries in IdentifiableCredentials at tallying, is a straight-forward solution. However, in SKHS12 it is not that simple: If the entry is ignored prior to post-registration, $\mathcal{A}$ would expect the #-component of the real credential to be missing in UnlinkableCredentials. However, if he finds it to be there, he will know that #′ was not the correct #-component of $V^{\mathcal{A}_C}$'s credential. If the entry is only ignored prior to tallying, an illegitimate vote cast with $V^{\mathcal{A}_C}$'s retained credential **cred** would be eliminated only at the stage where M-PET is performed on the input of UnalteredVotes and UnlinkableAssignedσComponents. However, if $\mathcal{A}$ casts a vote using the fake credential **cred**′ he received from $V^{\mathcal{A}_C}$, he may find that no votes are eliminated at all when M-PET is performed. Thus again, he learns that he has received a wrong credential.

We therefore define credential-retention by having the registrars compute a new value $\tilde{\sigma}$ and replace the $E(\sigma)$-component of **Cred** on IDENTIFIABLECREDENTIALS with an encryption of $\tilde{\sigma}$. The encryption of the #-credential however remains the same. Finally, the voter's identifer associated with the entry on IDENTIFIABLECREDENTIALS is marked as *non-eligible*. The new credential in IDENTIFIABLECREDENTIALS is marked and may not be assigned to new voters, since the coercer would know the true value of the #-credential, in case it previously belonged to a voter controlled by him. Clearly, voters that loose their right to vote will not be able to use their retained credential for voting, since such votes would be discarded at step 4 of tallying.

Recall that the JCJ-model for assessing coercion-resistance assumes exactly one voter $V^{\mathcal{A}_C}$ who can potentially be subject to vote-buying or coercion. However, it seems that a scheme that addresses these concerns should be suitable to protect a larger fraction of the electorate. Indeed, in JCJ and SKHS11 all voters are protected against coercive attacks regardless of their number, as long as they may assume a sufficiently large set $\mathcal{V}^{\mathcal{H}}$ of honest voters. In fact, the extra votes cast due to other voters being coerced, serve as noise votes for the individual voters under coercion. In SKHS12, this does not hold. Since coerced voters *use up* the #-components of UNLINKABLE#CREDENTIALS, it becomes increasingly likely that the individual voters choose $\#'$ as a value that is already used by another voter. Thus, the adversary's chances at detecting a defense strategy by observing CASTVOTES for colliding #-components increase. We introduce the notion of δ_{multi} to capture the case where multiple voters are under coercion. It is computed the same way as δ. However, we change the definition of the underlying coercion games, which now allow a fraction $a > 0$ of N voters to be coerced instead of just one. It is easy to see that $\delta_{multi} = \frac{N_+ \cdot (1+a) - 2}{\beta \cdot N_+ - 1}$ in SKHS12, given the conservative assumption that all other voters choose an unused value as $\#'$. When further assuming the extreme case of $a = 1$, still we find that $\delta_{multi} = 2\delta$. In order to protect multiple voters from coercion in this extreme case, β would need to be roughly (less than) doubled. Thus, each of them is protected to the same degree as $V^{\mathcal{A}_C}$, when no other voters are coerced. The impact on the performance is reduced merely to the phases of pre-registration and post-registration - in the tables above, each of these values increases by 1.

Recall that abstainees enjoy full coercion-resistance in both JCJ and SKHS11. However in SKHS12 this is not the case, i.e. the adversary can in any case identify whether $V^{\mathcal{A}_C}$ has applied his defense strategy with a probability of δ. $V^{\mathcal{A}_C}$ cannot benefit from not wanting to cast a vote in the first place. On the positive side, even abstainees can verify that their right to vote has not been misused.

3.5 Related Schemes

In 2005, JCJ introduced a very strong notion of coercion-resistance at the cost of very weak trust assumptions. However, JCJ comes with high computational costs, particularly during the phase of tallying. With the two schemes presented above, we aimed at reducing this problem while staying as loyal as possible to the JCJ premises. In the following two sections, we shortly introduce other schemes that improve efficiency.

Particularly, we will look into SHKS11, Selections and KHF11 and relate them to the protocols presented above.

There are two further schemes that have been proposed recently. The scheme presented in [7] is an improvement of the well-known scheme by Araújo et al. from [6] and [4]. It relies on the q-Strong Diffie-Hellman and the Strong Diffie-Hellman Inversion assumptions. Indeed, the protocol is efficient in all phases while providing coercion-resistance with $\delta = 0$. The first version of the protocol was flawed, since eligibility verifiability was not granted and credential retention could not be done. Later in [7], Araújo et al. manage to fix these issues. In [29], Essex et al. present a scheme that allows to authorize votes and establish the list ValidVotes already during the phase of vote casting. This renders the tallying phase outstandingly efficient - all that remains to do is to decrypt and count the votes. However, a high price is payed during registration, where each voter needs to perform a vast amount of computations. Also, the credentials cannot be reused at subsequent voting events without comprimising coercion-resistance. However, the scheme seems to be parametrizable to achieve a level of coercion-resistance where $\delta = 0$ if using the credentials only once.

Before discussing the remaining three schemes, we note that there exist practical non-efficiency related issues in coercion-resistant Internet voting that need to be explored. Crucial aspects, for instance how voters can obtain and manage their (real and fake) credentials and by which means they should be able to benefit from verifiability remains unanswered in JCJ. However, work has been done to address these issues. In [19] a protocol is defined that in parts modifies the original JCJ to address the efficiency shortcomings. An implementation of JCJ has been made based on this work. In [64] and [65] solutions to these problems are shown based on smart-cards. [55] and [54] show ways of how voters can manage their credentials, **cred** and **cred**$'$.

3.5.1 SHKS11 Protocol and CH11 Protocol (Selections)

Although technically the schemes are quite different, both use a similar approach to achieve the uncertainty related to coercion-resistance. As in the protocols introduced above, they involve a parameter β which determines the tradeoff between the dergree of coercion-resistance δ and efficiency. When casting a vote, the voters assign it to their public credential **Cred**, which in return is linked to their identity. In order to provide coercion-resistance, the votes are assigned not only to the identity of the actual voter, but also to a set of $\beta - 1$ additional voters, thus obtaining an *anonymity set* of β voters.

After ruling out duplicates in SHKS11 [74], β tuples are formed. Each tuple contains an identical copy of the information cast by the voter $(E(\hat{\sigma}), E(v))$ and a distinct public credential $E(\sigma_i)$ as it appears in IdentifiableCredentials of a voter V_i in the anonymity set. Before performing PET to remove unauthentic votes, the tuples are processed by a mix-net. Clearly, the tallying phase scales linearly in β.

In Selections [17], voters need to prove that the encrypted credential $E(\sigma)$ cast along with their vote and other values (we omit the details) is a re-encryption of one out of β credentials on $\mathcal{PB}$. The size of the $\mathcal{OR}$-proof the voters need to furnish scales linearly in β.

In [83] the relation between β and δ is observed. The adversary's strategy to detect whether $V^{\mathcal{A}_C}$ applied his defense strategy lies in counting the number r of times a vote was assigned to his identity. Given the definition from section 3.1.3, it is easy to see that $\delta = \sum_{\delta_r > 0} \delta_r$, where $\delta_r = F_{bin}(r, n_{cast}, \frac{\beta-1}{N-1}) - F_{bin}(r-1, n_{cast}-1, \frac{\beta-1}{N-1})$. The function $F_{bin}(r, t, p)$ is the distribution function of a binomial distribution with r successes, t trials and success-probability p. Although it seems that for not too small values of δ, the schemes are quite efficient, β generally needs to be chosen rather big in order to achieve high degrees of coercion-resistance. The following table shows the smallest value of β to achieve or exceed a given degree of coercion-resistance δ with a number of N voters. We assume that the number of cast votes n_{cast} equals the number of voters N.

	$\delta = 1/10$	$\delta = 1/20$	$\delta = 1/50$	$\delta = 1/100$	$\delta = 1/200$
$N = 1000$	$\beta = 17$	$\beta = 61$	$\beta = 286$	$\beta = 615$	$\beta = 865$
$N = 100'000$	$\beta = 17$	$\beta = 65$	$\beta = 398$	$\beta = 1586$	$\beta = 5987$

The following table holds the number of modular exponentiations needed for the fourth step at tallying, i.e. *remove unauthentic votes* in SHKS11. The same values of δ and N are observed as in section 3.4.4. Again, we show the next higher integer to the logarithm in base 2.

	$\delta = 1/20$	$\delta = 1/100$
$N = 1000$	21	24
$N = 100'000$	27	32

SHKS11 performs particularly well for large N and not too small δ. A great asset of this protocol however is its striking simplicity. In the next table we perform the same analysis for the vote casting step of Selections.

	$\delta = 1/20$	$\delta = 1/100$
$N = 1000$	9	12
$N = 100'000$	9	13

As SHKS11, Selections performs particularly well for not too small δ. For higher degrees of coercion-resistance, the capacities of home devices may reach their limits when putting the protocol to practice. It is worthwile mentioning a special feature specific to this scheme. With many other protocols, inlcluding JCJ, voters need to cast a vote in order to verify that no other vote has been cast using their credential **cred**. In Selections, voters who want to abstain can do so and yet verify that their right to vote has not been misused. The same is true for the schemes in [40], [7] and [17]. In terms of protocols, this feature merely protects against a collusion of registrars or talliers. However in practice, voters may find it hard to keep their credentials secret. This feature therefore seems to be particularly beneficial for practice.

3.5.2 KHF11 Protocol

In [53], Koenig et al. introduce a protocol that primarily aims at excluding *board-flooding attacks*. By casting large amounts of unauthentic votes to $\mathcal{PB}$, in JCJ attackers can virtually flood the list CastVotes and render tallying unbearably long. The proposal in [53] manages to limit the number of votes in CastVotes, without excluding any valid votes and of course without waiving coercion-resistance. Remarkably, the scheme is very efficient at both vote casting and tallying as well. Furthermore, both steps are insensitive to the desired degree of coercion-resistance. Similarly as in SKHS12, only preparation steps are affected. If these processes are started early enough, they will not have any negative impact on the voting operations.

After registration each voter V_i owns a number $d_i - 1$ of dummy credentials apart from his real credential σ_i. Whenever the adversary $\mathcal{A}$ tells V_i to hand out σ_i, he hands out one of the dummy credentials instead. Votes that are not cast using either σ or a dummy credential are immediately excluded from further processing. Thus, the timespan of voting is used for excluding a potantially large number of unauthentic votes at an early stage, rather than waiting for the polls to close. In the end a maximum of $\sum_{i=1}^{N} d_i$ votes reach the tallying stage. Clearly, for coercion-resistance it is crucial that $V^{\mathcal{A}_C}$ can lie about the number of dummy credetials he obtained. [55] and [54] show how he can manage them while keeping their number secret. The only drawback regarding the degree of coercion-resistance lies in the ability of $\mathcal{A}$ to obtain the actual credential σ with a non-negligilbe probability. Except for SHKS12, in the previously discussed protocols $V^{\mathcal{A}_C}$ enjoys maximum protection from coercers and vote-buyers, i.e. $\delta = 0$, as long as he wishes abstain from casting his own vote (line 7 in the real coercion game).

[40] generalizes this approach to a generic scheme, where each voter receives d_i *posting tickets*. Each time voters cast a vote to $\mathcal{PB}$, they use up a ticket. However, the tickets can be re-used at subsequent votes without compromising coercion-resistance in any way. The scheme can be built on top of any protocol and avoid board-flooding. The computation time of the preparatory steps is linear in the overall number of tickets.

Clearly, the degree of coercion-resistance hinges on the number of dummy votes, as does the efficiency of the preparatory steps. [40] discusses a possible distribution function $f^*(\cdot)$ according to which d_i could be distributed, where $f^*(\cdot) : \mathbb{N} \to [0, 1]$ is a discrete distribution function that essentially interpolates to a shifted density function f of a normal distribution. Clearly, $f^*(0) = 0$, since otherwise some voters could not cast any vote at all. Also, there may exist no a such that $f^*(a) = 1$, since otherwise the scheme were not coercion-resistant to any extent, i.e. $\delta = 1$.

It is argued that the degree of coercion-resistance is $f^*(1)$. The reasoning assumes an adversary $\mathcal{A}$ who asks the voters to hand out their tickets. He accepts a run when receiving at least one. Clearly, the probability of receiving at least one credential is 1, given that $V^{\mathcal{A}_C}$ complies with $\mathcal{A}$'s demand. The probability of receiving at least one ticket, given that $V^{\mathcal{A}_C}$ applies his defense strategy, is $\sum_{k=2}^{\infty} f^*(k)$. The difference between these two values is $f^*(1)$, i.e. $\delta = f^*(1)$.

However, we believe that this reasoning makes an assumption on $\mathcal{A}$'s capabilities, which may seem natural in certain settings but not necessarily in the general case. Particularly, if $\mathcal{A}$ is a coercer who becomes violent in case of receiving no ticket from $V^{\mathcal{A}_C}$, $\mathcal{A}$ may find it imperative to give $V^{\mathcal{A}_C}$ a chance to avoid being punished, for the sake of his own ($\mathcal{A}$'s) credibility. This reasoning inevitably ends in the decision of accepting a run, even when getting one ticket only. Now, imagine $\mathcal{A}$ were a vote-buyer. Indeed, intuitively, vote-buyers seem to be more difficult to defeat than coercers, since complying voters have no interest in exposing them. While a coercer will find it hard to explain why he punishes somebody for not having received more than one ticket, a voter-buyer may call it a *part of the game* not to pay a voter in such a case.

We now observe how δ is computed due to this reasoning, i.e. we imagine $\mathcal{A}$ to be a vote-buyer rather than a coercer. For simplicity, we assume that σ_i is one of the d_i tickets $V^{\mathcal{A}_C}$ received. $\mathcal{A}$'s goal is to get as many σ_i as possible from a given subset of voters by spending at most the same amount of money he is willing to pay when using a system with $\delta = 1$. Clearly, he will define the rules such to yield δ as big as possible. His freedom lies in defining the number of tickets a for which he accepts a run. Each value of a corresponds with a possible degree of coercion-resistance δ_a. We find that $\delta_a = \sum_{k=a}^{\infty} f^*(k) - \sum_{k=a+1}^{\infty} f^*(k)$. Clearly, δ_a is greatest when choosing $a = k$, for any k that satisfies $f^*(k) = \max f^*(\cdot)$. The degree of coercion-resistance is therefore $\delta = \max f^*(\cdot)$ when considering a vote-buyer. The result is rather intuitive, since the vote-buyer thus motivates a maximum of voters to hand out their credential σ.

Given this result, evidently the optimal distribution function $f_{opt}(\cdot)$ yielding $\delta = \max f^*(\cdot)$, i.e. the one with the least number of tickets, is defined by $f_{opt}(k) = \delta$, for $0 < k \cdot \delta \leq 1$; $f_{opt}(k) = 1 - (k-1) \cdot \delta$, for $1 < k \cdot \delta \leq 1 + \delta$; $f_{opt}(k) = 0$ otherwise. If $\frac{1}{\delta}$ is an integer, $f_{opt}(\cdot)$ defines a uniform distribution over $[1, \frac{1}{\delta}]$. Each voter thus receives an average of $\frac{\delta+1}{2\delta}$ tickets, which is roughly half the amount of when using $f^*(\cdot)$. The following table holds the number of modular exponentiations needed for the preparatory steps in [53] when using $f_{opt}(\cdot)$. They consist of mixing the set of encrypted tickets and preparing the values to perform M-PET with the cast votes. We assume that the encrypted tickets are prepared prior to registration, in oder to allow the mixing and the preparation of M-PET to take place while voters register, similarly as in SKHS12. The preparatory steps however need to be finished before the phase of vote casting. Again, we show the logarithms in base 2 as above.

	$\delta = 1/20$	$\delta = 1/100$
$N = 1000$	18	20
$N = 100'000$	24	26

The performance is well in all cases. The scheme does not only manage to limit the input of votes to the tallying phase but it also manages to render the critical steps efficient, i.e. vote casting and tallying. A certain price is still payed during the preparation phase. However, this phase is less time-critical and scales only linearly in the average number of tickets per voter.

3.6 Conclusion

We have introduced two protocols that are δ-coercion-resistant in a parameter β. Both significantly reduce the running time of JCJ during the tallying phase. Yet, the efficiency in SKHS11 still scales linearly in the parameter β. Thus, if a high degree of coercion-resistance is desired, voters may still need to wait long before getting the results, particularly in the case of a large electorate. This is due to the potentially large number of noise votes. However, the scheme is strikingly simple and easy to explain. Further, it offers a lot of flexibility at finding the desired tradeoff between efficiency and the degree of coecion-resistance. While a ballot may start with none or few noise votes, it is simple to increase their number whenever rumours of coercion or vote-buying come up. Also the number of noise votes can be defined individually per voter. SKHS12 does not offer this kind of flexibility. However, it manages to render the tallying phase very efficient and independent of the parameter β. The price for coercion-resistance is only payed during the preparation phase. Thus, noone needs to be kept waiting at any stage. The problem of board flooding can be solved in both cases by combining with the approach presented in [40].

Apart from efficiency issues, the question remains how voters should manage their credentials. Particularly, voters are not able to memorize a credential $\mathbf{cred}$ that is as long and random as σ. However, work has been done to examine how this problem could be solved by using smartcards that access the credentials upon the entering of a password. In order to obtain a fake credential $\mathbf{cred}'$, another password would need to be entered. Clearly, the positive effect on coercion-resistance may be limited in practice when considering pressure that is applied by family members. Nevertheless, if vote-buying or other forms of remote bribery or coercion are a particular concern, the work done for instance in [55] provides foundations for solutions. Furthermore, it can be quite easily combined with the protocols discussed in this chapter.

Finally, we note that even if an efficient implementation can be found on paper, such a system is likely to be more costly in many terms than a more simple one. Indeed, the needs assessment may reveal that coercion and vote-buying are only limited concerns in a given country. In many cases it seems reasonable to believe that the authorities would hardly influence a ballot by buying votes at the risk of getting caught. If this is the case, more simple schemes can be proposed which rather focus on protecting voters from actual third-parties, i.e. not from the authorities. Particularly, restricting universal verification to a trusted group may already solve the problem, especially in countries where people usually vote by mail, as in Switzerland. In the Norwegian 2011 and 2013 trials, voters were allowed to cast multiple votes, whereas the last one counted and paper votes always overruled electronic votes [35]. Thus, the concern of family voting was addressed. The next chapter elaborates on how coercion-resistance can benefit from integrating the Internet channel with the polling-station.

Chapter 4

More Efficiency Thanks to Hybrid Schemes

This chapter contains the results from four peer-reviewed publications [81], [80], [41] and [26]. They were written under strong participation of the author of this thesis. The following parts of this chapter contain some extracts from these papers with only few changes.

The previous chapter has shown ways to achieve efficient coercion-resistant Internet voting. Since they do not only offer coercion-resistance but also a strong sense of verifiability, such schemes are still difficult to put into practice, particularly in a user-friendly way. There is also another reason why these schemes may not become a first choice in political voting. Indeed, Internet voting will hardly replace the conventional voting channels. Voting at the polling-station and in some cases voting by mail will still be available for decades. Therefore, methods need to be put in place to exclude the event of voters casting multiple votes, e.g. one through the Internet and one at the polling-station. Since coercion-resistant schemes hinge on the voters' ability to lie about having cast a vote, it remains challenging to enforce the one-man-one-vote principle efficiently under this condition. Obviously, there is no way to do so without compromising coercion-resistance and/or verifiability in some way.

The work in this chapter addresses coercion-resistance with regard to the situation, where voting at the polling-station and voting through the Internet are provided in parallel. Recall, that receipt-freeness is a strict condition to coercion-resistance in pure Internet voting. Interestingly, the presence of voting receipts, i.e. information a voters can use to unambiguously reveal to third-parties how they voted, now become a welcome instrument at providing coercion-resistance.

In section 4.1 we introduce our notion of a *hybrid scheme*, which entails an Internet voting channel and voting at the polling-station. The aim is to enforce the one-man-one-vote principle and coerion-resistance of the overall scheme and verifiability of Internet voting. We show which types of protcols can be used for Internet voting and which ones are not suitable. Section 4.2 shows a protocol that is strikingly simple and meets

the requirements for an Internet voting protocol within a hybrid scheme. Finally, in section 4.3 we describe how the protocol could be implemented as a proof of concept. A simplified version was put into practice and used within the *Baloti* project, which we will also shortly introduce. Also, it was used in succeeding projects, for instance at providing student board elections of Swiss universities [27].

4.1 Hybrid Schemes

In the following section we introduce the principles of hybrid schemes. The basic ideas have first been proposed in [78]. In section 4.1.2, we present which classes of protocols are suitable for the Internet channel of a hybrid scheme. Depending on the class a selected protocol belongs to, we show two different ways of letting voters revoke their electronic vote.

4.1.1 Principles

A hybrid voting scheme offers the choice between casting a vote through the Internet or paying a visit to the polling station. Vote-buying and coercion are undermined by allowing the voters to revoke their electronic votes at the polling station. Afterwards they are free to cast another vote inside the polling station, i.e. in a controlled and presumably coercion-free environment. We do not address voting by mail, however the polling station procedures can easily be enhanced to accomodate remote voting on paper as well. However, in these cases the concerns regarding coercion and vote-buying are likely to be much lower in countries where voting by mail is common. Clearly, the revocation mechanism must be designed in a way that an adversary can not find out which votes have been revoked. In 4.1.2, we will propose two different solutions to that problem. Both solutions include three different ballot-boxes: the α-*box* for the electronic votes, the β-*box* for the vote revocations, and the γ-*box* for the paper votes. The final outcome Σ of the voting can then be calculated as

$$\Sigma = \alpha - \beta + \gamma,$$

where α, β, γ denote the individual results of the respective ballot-boxes. The results are computed for each voting choice $c \in \mathcal{C}$. Depending on the revocation mechanism, the β-box may contain revocations either in electronic form or on paper. Clearly, each vote in the β-box must reflect the corresponding vote from the α-box. If the α-box is operated within a verifiable system based on a $\mathcal{PB}$, which we assume, revoked votes cannot simply be removed, otherwise coercion-resistance would be compromised.

In a hybrid scheme, adversaries must always assume that votes might be overruled by the voter's personal choice. Thus, even if an adversary is convinced that the voter cast the electronic vote to the α-box as told, he can never be sure that it is the one that will count in the end. Clearly, by witnessing the voter enter the polling station, the coercer would most likely suspect that his intention is to revoke his vote and not to play tennis. However, we do not aim at improving the polling-station procedures despite

their inevitable flaws. The aim is rather to benefit from their qualities and bring them to their best, i.e. for the sake of Internet voting as well.

Remarkably, Internet voting schemes do not necessarily need to fulfill the same requirements if they are employed within a hybrid system. For example, the electronic channel of a hybrid system does not need to provide receipt-freeness. On the contrary - receipts may even be an asset within a hybrid system. One of the proposed methods in section 4.1.2 even requires *guaranteed receipts*. However generally, we can be less restrictive. The following minimal requirements need to be met by the Internet channel of a hybrid system:

1. It needs to tell the polling-station staff whether a voter in the polling-station has cast an electronic vote.

2. In case a voter has cast an electronic vote, it needs to provide the polling-station with the encrypted vote or the vote itself.

When considering verifiable schemes, all the information needs to be proved correct, since we only want to rely on the trustworthiness of the polling-station staff. Due to the first requirement, the electronic channel needs to provide a *proof of eligibility*. A proof that a voter has cast a particular vote due to the second requirement we call a *receipt*. A proof that a voter has cast a particular ciphertext (possibly with a different randomization) as his vote we call a *vote identifer*. In order to obtain the information due to the two requirements, possibly the voter needs to bring along cryptographic material to the polling-station. This depends on the chosen implementation.

Note, that the existence of a mechanism to check if somebody has already voted electronically (first requirement) does generally not allow to identify that person's vote in the α-box (second requirement), because the system may provide a list of voters that is completely disconnected from the list of votes. Similarly, the guaranteed existence of a receipt (second requirement) may be insufficient for the staff to verify whether someone has cast an electronic vote or not (first requirement). Clearly, since receipts may only be known to the voter, it is easy for him to withhold it. However, as the most simple solution, both requirements can be met by leaving the encrypted vote attached to information that publicly identifies the voter.[1] Thus, he does not need to bring along any information due to the above requirements, which makes it simpler for him too. We conclude that the requirements can actually be met quite easily.

4.1.2 Revocation Mechanisms

To prevent vote-buying and coercion, we need to define a secure vote revocation mechanism that allows voters to revoke and replace their electronic votes at the polling-station.

[1] [81] contains the following statement, which we find to be misleading: *In order to preserve the voters' privacy, the individual votes clearly may never be decrypted in this case, not even at the time of tallying. Instead, homomorphic methods for tallying exist, where only the result of the tally needs to be decrypted.* Clearly, a mix-net could also be applied prior to decryption. Refer to the final state of the example protocol in chapter 2.

We only consider protocols that are verifiable by the means of a $\mathcal{PB}$. What we previously called the α-box is in fact a designated area on the $\mathcal{PB}$, particularly the list CastVotes. In order to revoke a vote, it should not be necessary to apply the decryption key held by the tallying authorities.

The traditional voting infrastructure needs to satisfy the following three minimal requirements.

1. The traditional voting infrastructure consists of a polling station, where the paper votes of registered voters are anonymously collected in a physical ballot-box (the γ-box).

2. The traditional voting procedure at the polling station (checking the identity of voters, opening the ballot-box, counting the votes, etc.) is sufficiently secure, in particular coercion-resistant, and the group of voting officials is reliable and trustworthy.

3. The official voting period at the polling station chronologically succeeds the electronic voting period.

To understand the applicability of the proposed vote revocation procedures, we first need to get an overview of the different types of electronic ballot-boxes in Internet voting protocols. Depending on the chosen configuration and properties of $\mathcal{PB}$ and the structure of its entries, Internet voting systems can be classified into the follwing three categories:

1. The Internet voting system guarantees a receipt that is constructable by the voter alone or in collaboration with the polling station staff. (Example: SH10 introduced in the following section.)

2. The Internet voting system guarantees a vote identifier that is constructable by each the voter or the polling station staff alone or in collaboration with eachother. (Examples: SH10 and HS11 introduced in the following section, as well as the example protocol presented in 2.)

3. The Internet voting system guarantees neither a receipt nor a vote identifier. (Examples: The coercion-resistant protocols presented in chapter 3.)

Procedure 1: Revocations on Paper

The first procedure we propose assumes the guaranteed presence of a receipt for any given vote in the α-box. The payoff of this restriction is a revocation procedure that particularly appeals by its simplicity. The following points define the procedure. We start off when the voter at the polling station is about to revoke the electronic vote in the α-box (we assume that the voting offcials have already successfully checked the voter's right to vote and his proof of eligibility).

1. The voter (possibly in collaboration with the staff at the polling station) reveals the receipt for his vote in the α-box towards the voting officials.

2. The voting officials prepare a revocation paper ballot containing the same vote and hand it over to the voter.

3. The voting officials verify that the voter drops the revocation paper ballot into the β-box.

4. The voter is granted access to the γ-box to cast the final paper vote.

In this procedure, the β-box is a physical ballot-box similar to the γ-box. At the end of the official voting period, it is opened and tallied according to the same procedure.

Note, that in the scheme as it is proposed, it is crucial to assume that the voting officials will not allow the voters to cast a paper ballot that differs from their electronic votes in the α-box. If not all voting officials are considered to be fully trustworthy, then several voting officials should be involved in each step of the procedure. In other words, before the voter gets access to the γ-box, a sufficient number of voting officials would have to give their approval, for instance by signing the revocation ballot. Thus, we merely need to assume that among the group of involved voting officials, there is at least one that would refuse the signature to an incorrect revocation ballot.

A drawback of this procedure is the fact that the content of the electronic vote must be revealed to the voting officials. One could argue that this violates the secrecy requirement, because in a simple yes/no-type of voting, one could guess that revoking a yes-vote implies that the update will be a no-vote, and vice versa. On the positive side, the procedure allows coercion-resistance even if the space of voting options $\mathcal{C}$ is large. Particularly, an adversary who aims at launching an attack based on instructing a voter to cast a highly unprobable vote (*Italian attack* [22]), will not be able to verify his compliance by observing $\mathcal{PB}$.

Procedure 2: Electronic Revocations

Let the Internet voting component of the hybrid system now be a system that provides a mere vote identifier, not necessarily a receipt. The idea then is to leave the votes encrypted throughout the whole revocation procedure. To guarantee the anonymity of those who decide to revoke their votes, and thus to ensure the overall system to remain coercion-resistant, we define the β-box as a section of $\mathcal{PB}$ to which re-encryptions of the original votes are posted. Clearly, the votes may not be linked to any information that identifies the voters. The adversary is then unable to make out which votes from the α-box have been revoked. The encryption scheme used to generate the encrypted votes in the α-box must allow re-encryption and the generation of a non-transferable proof of correct re-encryption. This requirement is met by ElGamal introduced in section 2.5. Provenly correct re-encryptions are often done by mix-nets as introduced in section 2.8.

The procedure is defined as follows:

1. The voter generates a re-encryption of the encrypted vote in the α-box.

2. A corresponding non-transferable proof of correct re-encryption is generated, designated to the voting officials at the polling station. Optionally, this step can be done remotely in a non-interactive manner, given the existence of trusted software.

3. The voter approaches the voting officials and uses the vote identifier to identify the encrypted vote in the α-box.

4. The voter hands the re-encryption and the corresponding non-transferable proof to the voting officials.

5. If the proof is accepted, the voting officials post the re-encrypted vote to the β-box.

6. The voter is granted access to the γ-box to cast the final paper vote.

Similar to the previous procedure, we can enhance it by requiring a sufficient number of voting officials to approve the voter's re-encryption: A voter would only be granted access to the γ-box once a sufficient number of voting oficials have posted their electronic signature of the re-encryption to the bulletin board. Clearly, the randomization factor the voter used for his re-encryption serves him as a receipt; He can always prove to an adversary that he has revoked his electronic vote. However, he will never be interested in doing so. On the other hand, the receipt does not help at proving to an adversary that he did not revoke his vote. It thus does not reduce the security level of the overall system.

4.2 Protocols for the Internet Channel - SH10 and HS11

The protocols SH10 [80] and HS11 [41] are both suitable as the Internet channel of a hybrid system. Unlike the example protocol in chapter 2, they both offer anonymity, given a trusted majority of trustees. Thus, fairness is not put at stake due to observing which political groups have managed to mobilise their electorate. Apart from individual verifiability, a particularly high level of universal verifiability is provided, where the public can verify that no collusion of trustees have used unassigned credentials for casting unauthentic votes. Further, abstainees can verify that their right to vote has not been misused, without requiring any trust in the trustees, unlike in JCJ. While SH10 can be used with both revocation procedures presented in the previous section, HS11 is restricted to the second one, since the presence of a receipt cannot be guaranteed. HS11 is an enhancement of SH10 aiming at being more easy to implement, i.e. with voters' devices that are limited in their computational performance. The core element of the protocol, which is essentially an anonymous shuffle of DSA public keys[2], has been proposed in [61] beforehand.

[2]DSA stands for the Digital Signature Algorithm, which is a widely known standard. We do not explain it in detail here. However, we emphasise that the public and private keys are computed the same way as in the ElGamal cryptosystem.

4.2.1 Protocol Overview

Before giving a more detailed description in the next section, we start off by presenting the main aspects.

- *Generation of Public and Secret Credentials:* As a precondition to a voting process, the protocol assumes the existence of a publicly readable voter-roll. It can be thought of as a list that identifies all eligible voters. Each voter V_i is assigned a *public credential* $\mathbf{Cred_i}$ and the matching *secret credential* $\mathbf{cred_i}$. The latter is kept secret by the voter. These values can be reused across multiple voting events. A voter's public credential is associated with his entry in the voter-roll and published as the set IDENTIFIABLECREDENTIALS. Without disclosing it, voters can prove that they own the secret credential that matches their public credential with a signture (a non-interactive Σ-proof is used in SH10, a DSA-signature is used in HS11). On the other hand, it is computationally infeasible to calculate the secret credential that matches a voter's public credential.

- *Generation of Pseudonyms:* Given the set IDENTIFIABLECREDENTIALS as input, a publicly readable set of shuffled pseudonyms UNLINKABLECREDENTIALS is generated before every voting event. Similarly as with public credentials, voters can prove that they own the secret credential that matches their pseudonym. On the other hand, it is computationally infeasible to calculate the secret credential that matches a voter's pseudonym. Associating public credentials from IDENTIFIABLECREDENTIALS with their corresponding pseudonym in UNLINKABLECREDENTIALS is computationally only feasible when knowing the corresponding secret credential $\mathbf{cred_i}$.

- *Vote Casting:* Voters use their secret credential $\mathbf{cred_i}$ and public values to compute their pseudonym and a signature of their vote. The signature can only be verified usng the pseudonym, i.e. not the public credential $\mathbf{Cred_i}$ linked with the voters' identities. Clearly, only voters who know $\mathbf{cred_i}$ are able to compute a signature that matches their pseudonym. In SH10 they also use $\mathbf{cred_i}$ to compute the encryption of their vote. The pseudonym, the encrypted vote, and the signature are posted to $\mathcal{PB}$ through an anonymous channel. If the proof holds against the sent values and if the supplied pseudonym is an element of UNLINKABLECREDENTIALS, the vote is considered authentic. By associating their vote with their pseudonyms, which is only possible when knowing the corresponding secret credential, voters authenticate themselves as eligible voters without disclosing their identity.[3]

- *Proofs of Eligibility and Vote Identifier:* As described in the previous section, the protocol must enable voters to prove that they have not cast an electronic vote. If they have cast an electronic vote, they at least must be able to reveal the vote they have cast and prove having done so correctly. Both requirements are satisfied by the knowledge of their secret credential $\mathbf{cred_i}$. At the polling station, voters

[3] In the literature, this concept is sometimes called *anonymous authentication* [73, 72].

authenticate themselves and identify their public credential in IDENTIFIABLECREDENTIALS. Further, they reveal the pseudonym in UNLINKABLECREDENTIALS that corresponds with their public credential and present a zero-knowledge proof to show that they have presented the correct pseudonym.[4] They can only do so using their secret credential. If there is no vote associated with that pseudonym, voters have proven their eligibility to cast their vote using the traditional paper-based infrastructure without prior revocation. If there is a vote associated with the pseudonym, the voter has proven ownership of that vote. In order to cast another vote using the paper-based infrastructure, it must first be revoked by following one of the revocation procedures described in the previous section (HS11 is restricted to the second procedure).

4.2.2 Detailed Protocol Definition

We divide the protocol into seven different steps, of which the first two do not need to be repeated at every voting event.

Step 1 - Setup. The protocol involves four groups of players, each of which is responsible for designated tasks as described in the following paragraphs.

1. Eligible *voters* $\mathcal{V} = \{V_1, .., V_N\}$.
2. *Registrars* $\mathcal{R} = \{R_1, .., R_{N_R}\}$.
3. *Pseudonym producers* $\mathcal{P} = \{P_1, .., P_{N_P}\}$.
4. *Talliers* $\mathcal{T} = \{T_1, .., T_{N_T}\}$.[5]

The registrars, pseudonym producers, and talliers are subgroups of trustees. Any intersection of groups can be non-void. Particularly, voters can work as registrars, pseudonym producers, or talliers at the same time. Regarding the secrecy requirements and anonymity, we require a trustworthy majority of members in each subgroup, whereas, as with mix-nets, at least one trustworthy pseudonym producer ist also sufficient. Regarding verifiability, none of the trustees need to be trusted.

The trustees agree on a generator g of a subgroup $\mathbb{G}_q \subset \mathbb{Z}_p^*$ of order q, such that p and $q = (p-1)/k$ are large primes, as if setting up a PKI for ElGamal as shown in section 2.5. These values are used across multiple voting events.

We further assume the existence of a voter-roll, an initially empty set IDENTIFIABLECREDENTIALS on $\mathcal{PB}$ and an anonymous channel for casting the votes.

Step 2 - Generation of Public and Secret Credentials. Objective: V_i knows his secret credential $\mathbf{cred_i}$ and the corresponding public credential $\mathbf{Cred_i}$ is published on

[4]Simply revealing the credential would compensate for the zero-knowledge proof. However, in that case voters would need to be assigned a new pair of public and secret credentials to meet the privacy requirement in subsequent voting events.

[5]Here we do not use this notation for the full group of trustees, but just the subgroup of talliers instead.

IDENTIFIABLECREDENTIALS along with an identifier from the voter-roll, such as name, birthday and address. These values can be reused across multiple voting events.

Definition. For each eligible voter $V_i \in \mathcal{V}$, the registrars $\mathcal{R}$ jointly create V_i's public and secret credentials using a distributed key generation protocol as proposed in [33]. The secret credential $\mathbf{cred_i}$ is a random value σ_i taken from $\mathbb{Z}_q$. His public credential $Cred_i$ takes the value g^{σ_i} and is published in the set INDENTIFIABLCREDENTIALS on $\mathcal{PB}$ associated with an identifier from the voter-roll. We will sometimes denote $\mathbf{Cred_i}$ as S_i for the sake of simplicity. Only a majority of $\mathcal{R}$ are able to compute σ_i. The members of $\mathcal{R}$ pass their shares of σ_i to V_i through a sufficiently secure channel. This could for example be done through the postal system or by V_i showing up at the registration offices for in-person authentication. The received shares allow V_i to efficiently compute σ_i. Note that unlike in the protocols shown in the previous chapter, no untappable channel is required, since V_i does not need to be able to lie about the true value of σ_i for the sake of coercion-resistance.

Step 3 - Generation of Pseudonyms. Objective: For every $V_i \in \mathcal{V}$, the pseudonym $\hat{S}_{\pi(i)} = \hat{g}^{\sigma_i}$ is published at position $\pi(i)$ in UNLINKABLECREDENTIALS on $\mathcal{PB}$. $\hat{g} \in \mathbb{Z}_p^*$ is the so-called *pseudonym generator* and π is an unknown permutation of $\{1, .., N\}$. This step is conducted prior to every voting event.

Definition. Define $g_0 := g$ and $\mathbf{S}_0 = (S_{0,1}, .., S_{0,N}) := (S_1, ..., S_N)$. Taking g_0 and $\mathbf{S}_0$ as input, P_1 is responsible for the creation and publishing of g_1 and $\mathbf{S}_1 = (S_{1,1}, .., S_{1,N})$ according to the details given below. If the output of P_1 is verifiably correct, then P_2 uses it for the creation of g_2 and $\mathbf{S}_2 = (S_{2,1}, .., S_{2,N})$, and so on for all pseudonym producers $P_j \in \mathcal{P}$. At the end of the chain, P_{N_P} outputs the resulting pseudonym generator $\hat{g} := g_{N_P}$ and the permuted list of pseudonyms $\hat{\mathbf{S}} := \mathbf{S}_{N_P} = (S_{N_P,1}, .., S_{N_P,N})$, which contains V_i's pseudonym $\hat{S}_{\pi(i)} = S_{N_P,\pi(i)}$ at position $\pi(i)$. The permutation $\pi = \pi_n \circ \ldots \circ \pi_1$ is the result of a sequence of individual permutations π_j, where P_j is responsible for selecting π_j. In the ideal case, in which all N_P pseudonym producers publish verifiably correct outputs, we obtain thus the following two chains of public values on the public bulletin board:

- $g = g_0 \rightarrow g_1 \rightarrow g_2 \rightarrow \cdots \rightarrow g_n = \hat{g}$,
- $\mathbf{S} = \mathbf{S}_0 \rightarrow \mathbf{S}_1 \rightarrow \mathbf{S}_2 \rightarrow \cdots \rightarrow \mathbf{S}_n = \hat{\mathbf{S}}$.

To produce g_j and $\mathbf{S}_j$ from g_{j-1} and $\mathbf{S}_{j-1}$, respectively, P_j chooses $\alpha_j \in_R \mathbb{Z}_q$ and π_j uniformly at random to compute

- $g_j = g_{j-1}^{\alpha_j}$,
- $S_{j,\pi_j(i)} = S_{j-1,i}^{\alpha_j}$,

for all $i \in \{1, \ldots, N\}$. Obviously, this implies $\hat{g} = g^{\alpha_1 \cdots \alpha_{N_P}}$ and thus $\hat{S}_{\pi(i)} = S_i^{\alpha_1 \cdots \alpha_n} = (g^{\sigma_i})^{\alpha_1 \cdots \alpha_n} = (g^{\alpha_1 \cdots \alpha_n})^{s_i} = \hat{g}^{\sigma_i}$, which means that the pseudonyms are evidently generated as intended. Note that V_i can independently compute $\hat{S}_{\pi(i)} = \hat{g}^{\sigma_i}$ using the public pseudonym generator $\hat{g}$ and the secret credential σ_i.

To avoid that the pseudonym producers deviate from the protocol by not choosing the values α_j uniformly at random, we ask them to select α_j and publish $A_j = g^{\alpha_j}$ prior to the pseudonym generation process. Thus, value A_j serves as P_j's commitment to α_j.

Finally, to ensure that the output of each pseudonym producer $P_j \in \mathcal{P}$ is verifiably correct, it must be equipped with a corresponding zero-knowledge proof of correctness Z_j. This proof includes three components, one that proves conformity with the commitment A_j, one that proves the correct computation of g_j, and one that proves correct shuffling. Algorithm 6 shows all the details of what P_j needs to do (assuming that g_{j-1} and $\mathbf{S}_{j-1}$ are correct inputs).

Algorithm 6 Calculate g_j, $\mathbf{S}_j$, Z_j

Require: g_{j-1}, $\mathbf{S}_{j-1}$, α_j, A_j
  $g_j \leftarrow g_{j-1}^{\alpha_j}$
  $\pi_j \leftarrow$ random permutation of $\{1, \ldots, N\}$
  $\mathbf{S}_j \leftarrow$ initialize as N-ary vector
  for all $i = 1, \ldots, N$ **do**
    $S_{j,\pi_j(i)} \leftarrow S_{j-1,i}^{\alpha_j}$
  end for
  $Z_j \leftarrow ZKP[(\alpha_j) : (A_j = g^{\alpha_j}) \wedge (g_j = g_{j-1}^{\alpha_j}) \wedge (\bigwedge_{k=1}^{N} \bigvee_{i=1}^{N} S_{j,i} = S_{j-1,k}^{\alpha_j})]$
  Post g_j, $\mathbf{S}_j$, Z_j to $\mathcal{PB}$, keep α_j, π_j secret

For the sake of simplicity, we assumed in algorithm 6 that all previous pseudonym producers $P_1, \ldots, P_{j-1}$ have correctly fulfilled their tasks and that the input parameters g_{j-1} and $\mathbf{S}_{j-1}$ have thus been computed correctly from g_0 and $\mathbf{S}_0$. By withdrawing this assumption, i.e., by considering the situation where pseudonym producers do miscomputations, choose incorrect inputs, or produce any type of incorrect outputs, P_j would need to verify all existing proofs Z_1 to Z_{j-1} before executing algorithm 6. Then, instead of simply taking the outputs of P_{j-1} as input, P_j selects the greatest value $k < j$ such that correct proofs exist for P_k and all its predecessors. Additionally, P_j needs to check that every P_ℓ involved in the chain of correct proofs (i.e. from g_k and $\mathbf{S}_k$ back to g_0 and $\mathbf{S}_0$, respectively) has correctly followed this rule for selecting the input parameters. Note that the same selection rule must be applied at the end of the pseudonymization process for the selection of $\hat{g}$ and $\hat{\mathbf{S}}$ (instead of simply taking g_n and $\mathbf{S}_n$).

A problem of algorithm 6 in its simple description is the size of the involved proof, which grows quadratically with the number of voters. As a counter-measure, we may break up the input vector $\mathbf{S}_k$ (and thus $\mathbf{S}_j$) into $\frac{N}{b}$ sub-vectors of size b (suppose m is a multiple of b). Algorithm 6 can then process each of these sub-vectors individually. This reduces the size of the involved proofs and therefore the total running time of algorithm 6 from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \cdot b)$. As pointed out in [41], there are also more sophisticated approaches to more efficient mixing based on [61] and [63]. Also randomized partial checking could be used, as presented in section 2.8.

Step 4 - Key Generation for Vote Encryption and Tallying. Objective: For vote encryption and tallying, corresponding keys of a secure (t, n)-threshold ElGamal

cryptosystem are generated (refer to sections 2.5 and 2.7). The private key d is shared among the members of $\mathcal{T}$ and the corresponding public key e is published. This step is conducted prior to every voting event.

Definition. An appropriate protocol for secure distributed key generation based on *Shamir's Secret Sharing Scheme* [76] is proposed in [33]. Refer to section 2.7 for more details. To apply it in the context of our voting protocol, we need a second generator $h \in_R \mathbb{G}_q \setminus \{1\}$ of the same subgroup $\mathbb{G}_q$, which is jointly selected at random by the members of $\mathcal{T}$. At the end of the protocol, a public key $e = h^d \in \mathbb{G}_q$ is published. The corresponding private key $d \in_R \mathbb{Z}_q$ is shared among the members of $\mathcal{T}$ and can only be computed by a majority. Any smaller coalition has no advantage over a single adversary who tries to compute d from h and e without owning a share. Parties that deviate from the key generation protocol will be detected and disqualified by the others.

Step 5 - Vote Casting. Objective: An ElGamal encryption $E(v_i) = (x_i, y_i)$ is cast to $\mathcal{PB}$ along with a signature z_i that is verified against $\hat{S}_{\pi(i)}$.

Definition. Voter V_i calculates the pseudonym $\hat{S}_{\pi(i)} = \hat{g}^{\sigma_i}$ and the ElGamal encryption $E(v_i) = (x_i, y_i) = (h^{r_i}, v_i \cdot e^{r_i})$ of the vote v_i. In SH10 the randomness r_i is selected as σ_i and the signature z_i is a non-interactive zero-knowledge proof $ZKP[(\omega) : (\hat{S}_{\pi(i)} = \hat{g}^{\omega}) \wedge (x_i = h^{\omega})]$. In HS11, the randomness is a fresh random value from Z_q and the signature z_i is a DSA signature. V_i then posts $(\hat{S}_{\pi(i)}, E(v_i), z_i)$ to CastVotes through an anonymous channel. If several votes are cast under the same pseudonym, only one of them is kept according to some policy. Note that despite anonymity, eligibility verifiability is granted, since only members of the voter-roll are assigned a pseudonym. The randomness r_i is a guaranteed receipt only in SH10, since in HS11 the adversary could cast the vote on behalf of V_i and keep the receipt r_i to himself. In SH10 however, V_i can decrypt a vote cast by the adversary by using his secret credential σ_i.

Step 6 - Tallying. Objective: The result of the tally Σ is published and provably correct.

Definition. First, the entries in CastVotes are assessed regarding legitimacy. Votes v_i are published in LegitimateVotes, if $\hat{S}_{\pi(i)}$ is a valid pseudonym enlisted in $\hat{\mathbf{S}}$, if the verification of z_i yields *true* (thus it is an authentic vote) and if it is not a duplicate, i.e. if it is the one to be counted according to the imposed policy (generally the first or the last vote).[6] Based on LegitimateVotes, the voters can overrule their votes at the polling station. Afterwards, a majority of $\mathcal{T}$ publicly reveal their share of d. They exclude the spoiled ballots and count the valid votes to obtain the result Σ. Now anybody could efficiently calculate d using any set of a majority of shares and decrypt all cast votes to compute, hence verify the final outcome.

Step 7 - Revoking the Vote at the Polling Station. Objective: V_i is able to either prove not having cast a vote or to identify the encrypted vote $E(v_i) = (x_i, y_i)$ on LegitimateVotes. With regard to the possible application of the second revocation

[6] Note that in case the last vote should count, in SH10 it is crucial to exclude subsequent votes cast with the same proof z_i.

procedure described in the previous section, V_i may even be able to verifiably disclose v_i if necessary.

Definition. The voting officials identify the public credential S_i in IDENTIFIABLECREDENTIALS, as V_i authenticates at the polling station. V_i then reveals his pseudonym $\hat{S}_{\pi(i)}$ on UNLINKABLECREDENTIALS together with $ZKP[(s_i) : (S_i = g^{s_i}) \wedge (\hat{S}_{\pi(i)} = (\hat{g}^{s_i})]$ as a proof of correctness. If there is no vote associated with $\hat{S}_{\pi(i)}$ in LEGITIMATEVOTES , V_i has proven the eligibility to cast a paper vote. Otherwise, V_i is clearly the owner of the encrypted vote associated with $\hat{S}_{\pi(i)}$.

If the applied revocation procedure requires v_i to be revealed, in SH10 V_i can use the secret credential σ_i as a receipt to present $ZKP[(\omega) : (x_i = h^{\omega}) \wedge (\frac{y_i}{v_i} = e^{\omega})]$. This proves that v_i has been revealed truthfully. By previously handing out σ_i to a coercer, V_i might not know v_i, but it can easily be calculated as $\frac{y_i}{e^{\sigma_i}}$. Due to the zero-knowledge property of Σ-protocols, the credential σ_i can be reused for subsequent voting events.

4.2.3 Security Features

We briefly relate the protocol definition to the security requirements presented in section 2.2.

Accuracy. Given that votes reach $\mathcal{PB}$ in an unchanged state, the *integrity*, *completeness* and *soundness* requirements are trivially met by an appropriate definition of $\mathcal{PB}$ and the fact that votes can be decrypted by any observing party at tallying-time. However, since the anonymous channel ist not necessarily authentic, voters in SH10 are required to verify that their votes actually reach $\mathcal{PB}$ in an unchanged state and to react accordingly otherwise, i.e. by resending or even revoking their vote. In order to avoid these steps, the protocol could be enhanced by additionally furnishing a DSA signature as in HS11.

Democracy. Eligible voters are assigned a public credential and a pseudonym. By the definition of the pseudonym generation process, anybody can verify that each eligible voter is assigned his designated unique pseudonym correctly. Assigning pseudonyms to citizens not enlisted in the voter roll is clearly impossible. Thus the requirement *eligibility* is met. Since multiple votes are excluded at tallying, *uniqueness* is achieved.

Vote-Privacy, Fairness and Anonymity. The requirement *vote-privacy* is achieved if a vote cannot be linked back to its owner. By sending their vote through the anonymous channel and relating it to their pseudonym, voters anonymously authenticate as eligible voters without disclosing their identity. Thus, the votes are detached from the information on their senders. By associating the votes only with the pseudonym, also *anonymity* is granted. We hereby note, that offering anonymity allows for the remarkably efficient tallying procedure, which only requires modular exponentiations at the decryption of votes (in prcatice, the votes' legitimacy can already be assessed prior to the actual tallying phase as originally proposed in [80]). *Fairness* is achieved by having a majority of $\mathcal{T}$ ensure that the votes remain encrypted until the time of tallying. However in the case of SH10, when applying a last-vote-counts policy, fairness could suffer due

to using the same randomness (the secret credential σ_i) for each subsequent vote. Yet, this problem could easily be solved by requiring the vote $E(v_i)$ to be over-encrypted, possibly using a different crypto-system, such that only a majority of talliers are able to obtain $E(v_i)$.

Verifiability. Voter V_i can easily verify that his vote v_i is cast-as-intended and stored-as-cast by finding it in CastVotes on $\mathcal{PB}$. By obtaining the provably correct decryption key d, the voters have all the information it takes to perform all tallying steps on their own. The protocols are thus individually and universally verifiable.

Coercion-Resistance. The protocols meet the requirements imposed on the electronic sub-component of a hybrid voting system. The definition of a hybrid voting system directly yields *coercion-resistance*, as explained in section 4.1.

4.3 A Proof of Concept for the Electronic Channel of a Hybrid Scheme

The SH10 and HS11 protocols introduced in the previous section have been picked up and served as the foundation for the implementations Selectio Helvetica [42][26] and UniVote [27]. developed and hosted by the Bern University of Applied Sciences. In the meantime several student board elections of Swiss universities have been conducted based on HS11. The Selectio Helvetica system can be considered an intermediate state of development, where for instance $\mathcal{PB}$ was still missing. However, the basic operations were all in place. The *Baloti* project run by the Centre for Democratic Studies in Aarau used the system to allow migrants in Switzerland with no right to vote to express their opinion at Swiss poplular initiatives and referenda between 2010 and 2011 [26]. A few hundred participants have cast their vote at each voting event.

Apart from a user-friendly implementation, one particular challenge was the absence of a final voter-roll. Voters were supposed to be able to register even after the polls have opened. Since the project operated on a low-scale budget, no infrastructure could be put in place to grant for the voters' credentials to be transmitted in a highly secure way. Also, there were no technical aids such as smart-cards that would have allowed to securely store them. In order to consider these aspects in a conscious way, an extended version of the protocol was proposed based on the email adresses of the voters.

The extended protocol underlying the SH system involves two additional players. The vote organizer assesses the voter's right to vote. The voting provider acts as an intermediary among voters and trustees, and writes to $\mathcal{PB}$. Initially IdentifiableCredentials holds a sufficient number of public credentials $\mathbf{Cred_i}$ to accomodate all voters ever to participate. The voters obtain their secret credential $\mathbf{cred_i}$ as follows: A voter first asks the vote organizer to sign his e-mail address in order to confirm that he is enlisted in the voter-roll. The voter then sends the signed e-mail address to the voting provider. He in return associates the voter's e-mail address with an unused public credential from IdentifiableCredentials and sends registration credentials back to the voter by e-mail. The voter chooses a password and uses it to compute one desig-

nated hash code per trustee. These hash codes are sent to the trustees along with the registration credentials. The trustees verify the credentials and map the hash code to their share of the private signature key corresponding to the voter's public signature key. Whenever a trustee receives a request with a valid hash value, it replies with the share of the private credential $\mathbf{cred_i}$ mapped to it. Thus, if voters want to cast their vote, they only have to enter their password.

Clearly, the e-mail provider could easily choose his own password and use the registration credential to obtain $\mathbf{cred_i}$. However, the voter would notice that, since registration credentials are only valid once. Anonymity may be slightly compromised, since the trustees holding the shares of $\mathbf{cred_i}$ may suspect that a voter is about to cast his vote when requesting his credential. However, the voters have the freedom to query their credential only once and use it later without querying it again. It seems that this rather user-friendly solution essentially preserves the security features of the original protocol.

4.4 Conclusion

Coercion-resistance in Internet voting is hard to put into practice, especially when the restrictive trust assumptions introduced in section 2.3 need to be accounted for. Particularly it is difficult to satisfy verifiability along with coercion-resistance. However, when thinking of practice in political voting, there are other challenges that will appear more pertinent to many. Evidently Internet voting will not replace the conventional voting channels anytime soon. Thus, the question arises of how to integrate the new channel as to ensure that voters cast one vote at most. Integrating the two channels seems to be solvable without introducing any additional security threats. It can even be done while offering a high degree of verifiability, i.e. verifiability that even holds under the restrictive trust assumptions introduced in section 2.3. At the same time, it hardly takes any additional effort to achieve coercion-resistance in such a setting. In this sense we have defined hybrid schemes in section 4.1 and we have shown which types of protocols are suitable for the Internet channel. Indeed, many known protocols meet the requirements. Interestingly, the ones originally meant to provide coercion-resistance are the least suitable. In section 4.2 we have presented two protocols that seem to be interesting candidates. On one hand, they offer anonymity without compromising verifiability, even for abstainees. At the same time, they are very efficient at tallying, particularly when assuming that anonymous channels can be relied on. In the meantime, several student board elections have been hosted successfully by a system based on an extension of the HS11 protocol.

Chapter 5

Conclusion

Conventional voting entails procedures that have evolved over decades and that are simple to explain. It is easy to argue that carelessness and attempts of systematic fraud are addressed appropriately. While these procedures are indispensible, they are also time-consuming and expensive. In the long run, Internet voting may grant for significant savings. Yet, some extra efforts remain inevitable in order to evade irregularities and doubts among the public. In this thesis, we have observed how Internet voting can be made efficient, when requiring a very rigid sense of verifiability and coercion-resistance. Indeed, the previously known solutions would have compromised the savings expected from Internet voting. We have managed to show ways to overcome the long waiting time at tallying. We have compared our solutions with other valid proposals from the literature with regard to security and efficiency.

Our proposals in chapter 3 assume a remarkably hostile environment. Even if all players among the voting authorities collude in an attempt to bias an election outcome, still each individual voter could notice. Registered voters are only able to sell their votes if all players among the voting authorities collude. Although the efficiency issue is solved, the proposals are still hard to implement in a user-friendly way. By assuming trust in the polling-station environment in chapter 4, we managed to define protocols that are much simpler to implement than the previous ones. This assumption appears to be justified - if the staff at the polling-station is not trusted, most probably Internet voting would not need to be trusted either. The solutions from chapter 3 should therefore be considered a mere contribution towards a far-away setting, where voting on paper will be banned and coercion and bribery are a particular concern.

In practice, we strongly believe that Internet voting needs to provide sound evidence that it has not biased a final result. Also the secrecy requirements require special attention by distributing data among independent peers. There is not much point in narrowing down the requirements much further, since the actual needs will differ. While strong measures against coercion may be found critical in countries where people vote within booths, others may be more relaxed if voting by mail has become an accepted standard. Similarly, if there is a tradition of having the public observe the tallying procedures, one may find it compulsory to provide universal verifiability by using a public bulletin board. Otherwise, it may seem sufficient to restrict universal verifiability

to a trusted circle. No matter what the choices are, they will need to withstand debates.

Bibliography

[1] Guidelines on transparency of e-enabled elections. Tech. rep., Directorate general of democracy and politcal affairs, GGIS (2010) 5 E, Council of Europe (2010)

[2] Adida, B., Neff, C.A.: Ballot casting assurance. In: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop. pp. 7–7. EVT'06, USENIX Association, Berkeley, CA, USA (2006), http://dl.acm.org/citation.cfm?id=1251003.1251010

[3] Araujo, R.: On Remote and Voter-Verifiable Voting. Ph.D. thesis, Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany (2008)

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

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

[6] Araújo, R., Foulle, S., Traoré, J.: A practical and secure coercion-resistant scheme for internet voting. In: Towards Trustworthy Elections. pp. 330–342 (2010)

[7] Araújo, R., Traoré, J.: A practical coercion resistant voting scheme revisited. In: VOTE-ID. pp. 193–209 (2013)

[8] Bangerter, E.: Efficient zero knowledge proofs of knowledge for homomorphisms. Ph.D. thesis (2005)

[9] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on Computer and communications security. pp. 62–73. CCS '93, ACM, New York, NY, USA (1993), http://doi.acm.org/10.1145/168588.168596

[10] Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: STOC'94, 26th Annual ACM Symposium on Theory of Computing. pp. 544–553. Montréal, Canada (1994)

[11] Brands, S.: Rapid demonstration of linear relations connected by boolean operators. In: In EUROCRYPT '97. pp. 318–333. Springer Verlag (1997)

[12] Burmester, M., Magkos, E.: Towards secure and practical e-elections in the new era. In: Secure Electronic Voting, pp. 63–76 (2003)

[13] Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM 24(2), 84–88 (1981)

[14] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: CRYPTO. pp. 89–105 (1992)

[15] Chevallier-Mames, B., Fouque, P.A., Pointcheval, D., Stern, J., Traoré, J.: Towards trustworthy elections. chap. On some incompatible properties of voting schemes, pp. 191–199. Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=2167913.2167924

[16] Clark, J.: Democracy Enhancing Technologies: Toward deployable and incoercible E2E elections. Ph.D. thesis, University of Waterloo (2011)

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

[18] Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. Tech. Rep. TR 2007-2081, Department of Computer Science, Cornell University (2007)

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

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

[21] Damgård, I.: On electronic voting schemes. Lecture notes, Aarhus University, Computer Science Department (2006)

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

[23] Die Bundesbehörden der Schweizerischen Eidgenossenschaft: SR 161.116 Verordnung der BK über die elektronische Stimmabgabe. Systematische Rechtssammlung (2013)

[24] Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: Proceedings of the 13th conference on USENIX Security Symposium - Volume 13. pp. 21–21. SSYM'04, USENIX Association, Berkeley, CA, USA (2004), http://dl.acm.org/citation.cfm?id=1251375.1251396

[25] Dreier, J., Lafourcade, P., Lakhnech, Y.: A formal taxonomy of privacy in voting protocols. In: Communications (ICC), 2012 IEEE International Conference on. pp. 6710 –6715 (june 2012)

[26] Dubuis, E., Fischli, S., Haenni, R., Serdült, U., Spycher, O.: Selectio Helvetica: A verifiable remote e-voting system. In: CeDEM'11, Conference for E-Democracy and Open Government. Krems, Austria (2011)

[27] Dubuis, E., Fischli, S., Haenni, R., Hauser, S., Koenig, R.E., Locher, P., Ritter, J., von Bergen, P.: Verifizierbare Internet-Wahlen an Schweizer Hochschulen mit UniVote. In: Horbach, M. (ed.) GI-Jahrestagung. LNI, vol. 220, pp. 767–788. GI (2013), http://dblp.uni-trier.de/db/conf/gi/gi2013.html#DubuisFHHKLJB13

[28] Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. Information Theory, IEEE Transactions on 31(4), 469 – 472 (jul 1985)

[29] Essex, A., Clark, J., Hengartner, U.: Cobra: Toward concurrent ballot authorization for internet voting. EVT/WOTE (2012)

[30] Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Proceedings on Advances in cryptology—CRYPTO '86. pp. 186–194. Springer-Verlag, London, UK, UK (1987), http://dl.acm.org/citation.cfm?id=36664.36676

[31] Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Seberry, J., Zheng, Y. (eds.) ASIACRYPT'92, Workshop on the Theory and Application of Cryptographic Techniques. pp. 244–251. LNCS 718, Gold Coast, Australia (1992)

[32] Furukawa, J., Sako, K.: An efficient scheme for proving a shuffle. In: Kilian, J. (ed.) CRYPTO'01, 21st Annual International Cryptology Conference on Advances in Cryptology. pp. 368–387. LNCS 2139, Santa Barbara, USA (2001)

[33] Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. In: Stern, J. (ed.) EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques. pp. 295–310. LNCS 1592, Prague, Czech Republic (1999)

[34] Gharadaghy, R., Volkamer, M.: Verifiability in electronic voting - explanations for non security experts. In: Electronic Voting. pp. 151–162 (2010)

[35] Gjøsteen, K.: Analysis of an internet voting protocol. Cryptology ePrint Archive, Report 2010/380 (2010), http://eprint.iacr.org/

[36] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)

[37] Groth, J.: Non-interactive zero-knowledge arguments for voting. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS'05, 3th International Conference on Applied Cryptography and Network Security. pp. 467–482. LNCS 3531, New York, USA (2005)

[38] Groth, J.: A verifiable secret shuffle of homomorphic encryptions. Journal of Cryptology 23(4), 546–579 (2010)

[39] Haenni, R., Koenig, R., Fischli, S., Dubuis, E.: Trustvote: A hybrid e-voting system for large-scale elections. Tech. Rep. 6, Bern University of Applied Sciences (2009)

[40] Haenni, R., Koenig, R.E.: A generic approach to prevent board flooding attacks in coercion-resistant electronic voting schemes. Computers & Security 33, 59–69 (2013)

[41] Haenni, R., Spycher, O.: Secure Internet Voting on Limited Devices with Anonymized DSA Public Keys. In: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. pp. 8–8. EVT/WOTE'11, USENIX Association, Berkeley, CA, USA (2011), http://dl.acm.org/citation.cfm?id=2028012.2028020

[42] Hauser, S.: Selectio Helvetica. Master thesis, Bern University of Applied Sciences, Biel, Switzerland (2011)

[43] Heather, J., Ryan, P.Y.A., Teague, V.: Pretty good democracy for more expressive voting schemes. In: ESORICS. pp. 405–423 (2010)

[44] Hirt, M.: Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting. Ph.D. thesis, ETH Zürich, Switzerland (2001)

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

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

[47] Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In: Boneh, D. (ed.) SS'08, 11th USENIX Security Symposium. pp. 339–353. San Francisco, USA (2002)

[48] Jonker, H.L.: Security Matters: Privacy in Voting and Fairness in Digital Exchange. Ph.D. thesis, Eindhoven University of Technology and University of Luxembourg (2009)

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

[50] Karayumak, F., Olembo, M., Kauer, M., Volkamer, M.: Usability analysis of helios - an open source verifiable remote electronic voting system. In: Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE) (2011), http://tubiblio.ulb.tu-darmstadt.de/54387/

[51] Khazaei, S., Terelius, B., Wikström, D.: Cryptanalysis of a universally verifiable efficient re-encryption mixnet. In: Proceedings of the 2012 international conference on Electronic Voting Technology/Workshop on Trustworthy Elections. pp. 7–7. EVT/WOTE'12, USENIX Association, Berkeley, CA, USA (2012), http://dl.acm.org/citation.cfm?id=2372353.2372360

[52] Khazaei, S., Wikström, D.: Randomized partial checking revisited. In: Proceedings of the 13th international conference on Topics in Cryptology. pp. 115–128. CT-RSA'13, Springer-Verlag, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-36095-4_8

[53] Koenig, R., Haenni, R., Fischli, S.: Preventing board flooding attacks in coercion-resistant electronic voting schemes. In: Camenisch, J., Fischer-Hübner, S., Murayama, Y. (eds.) SEC'11, 26th IFIP International Information Security Conference. Lucerne, Switzerland (2011)

[54] Koenig, R.E.: ELECTRONIC VOTING OVER THE INTERNET The Boon and Bane of Modern E-Society. Ph.D. thesis, University of Fribourg (2013)

[55] Koenig, R.E., Haenni, R.: How to store some secrets. IACR Cryptology ePrint Archive 2012, 375 (2012)

[56] Krenn, S.: Bringing Zero-Knowledge Proofs of Knowledge to Practice. Ph.D. thesis (2012), http://books.google.ch/books?id=mHgimAEACAAJ

[57] Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion-resistance and its applications. In: Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium. pp. 122–136. CSF '10, IEEE Computer Society, Washington, DC, USA (2010), http://dx.doi.org/10.1109/CSF.2010.16

[58] Langer, L., Jonker, H., Pieters, W.: Anonymity and verifiability in voting: understanding (un)linkability. In: Proceedings of the 12th international conference on Information and communications security. pp. 296–310. ICICS'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1948352.1948380

[59] Langer, L., Schmidt, A., Volkamer, M., Buchmann, J.: Classifying privacy and verifiability requirements for electronic voting. In: Fischer, S., Maehle, E., Reischuk, R. (eds.) GI Jahrestagung. LNI, vol. 154, pp. 1837–1846. GI (2009), http://dblp.uni-trier.de/db/conf/gi/gi2009.html#LangerSVB09

[60] Lipmaa, H.: On the cca1-security of elgamal and damgård's elgamal. In: Proceedings of the 6th international conference on Information security and cryptology. pp. 18–35. Inscrypt'10, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2031933.2031936

[61] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Samarati, P. (ed.) CCS'01, 8th ACM Conference on Computer and Communications Security. pp. 116–125. Philadelphia, USA (2001)

[62] Neff, C.A.: Verifiable mixing (shuffling) of ElGamal pairs. Tech. rep., VoteHere, Inc. (2004)

[63] Neff, C.A.: Verifiable mixing (shuffling) of elgamal pairs. Tech. rep., In proceedings of PET '03, LNCS series (2003)

[64] Neumann, S., Volkamer, M.: Civitas and the real world: Problems and solutions from a practical point of view. In: Availability, Reliability and Security (ARES), 2012 Seventh International Conference on. pp. 180–185 (2012)

[65] Neumann, S., Feier, C., Volkamer, M., Koenig, R.: Towards A Practical JCJ / Civitas Implementation. In: In: Cryptology ePrint Archive: Report 2013/464, 2013. (Jul 2013)

[66] Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In: Proceedings of the 5th International Workshop on Security Protocols. pp. 25–35. Springer-Verlag, London, UK, UK (1998), http://dl.acm.org/citation.cfm?id=647215.720390

[67] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 17th international conference on Theory and application of cryptographic techniques. pp. 223–238. EUROCRYPT'99, Springer-Verlag, Berlin, Heidelberg (1999), http://dl.acm.org/citation.cfm?id=1756123.1756146

[68] Pedersen, T.P.: A threshold cryptosystem without a trusted party. In: Davies, D.W. (ed.) EUROCRYPT'91, Workshop on the Theory and Application of Cryptographic Techniques. LNCS 547, vol. 547, pp. 522–526. Brigthon, U.K. (1991)

[69] Peters, R.A.: A secure bulletoin board. Master's thesis, Eindhoven University of Technology (2005)

[70] Rivest, R.L., Smith, W.D.: Three voting protocols: Threeballot, vav, and twin. In: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology. pp. 16–16. EVT'07, USENIX Association, Berkeley, CA, USA (2007), http://dl.acm.org/citation.cfm?id=1323111.1323127

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

[72] Sako, K., Yonezawa, S., Teranishi, I.: Anonymous authentication: For privacy and security. NEC Journal of Advanced Technology 2(1), 79–83 (2005)

[73] Schechter, S., Parnell, T., Hartemink, A.: Anonymous authentication of membership in dynamic groups. In: Franklin, M.K. (ed.) FC'99, 3rd International Conference on Financial Cryptography. pp. 184–195. LNCS 1648, Anguilla, British West Indies (1999)

[74] Schläpfer, M., Haenni, R., Koenig, R., Spycher, O.: Efficient vote authorization in coercion-resistant internet voting. In: Kiayias, A., Lipmaa, H. (eds.) E-Voting and Identity, Lecture Notes in Computer Science, vol. 7187, pp. 71–88. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-32747-6_5

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

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

[77] Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (Nov 1979), http://doi.acm.org/10.1145/359168.359176

[78] Skripsky, J.: Minimal Models for Receipt-Free Voting. Semester project, ETH Zürich (2002)

[79] Smyth, B., Ryan, M., Kremer, S., Kourjieh, M.: Towards automatic analysis of election verifiability properties. In: Proceedings of the 2010 joint conference on Automated reasoning for security protocol analysis and issues in the theory of security. pp. 146–163. ARSPA-WITS'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1927614.1927625

[80] Spycher, O., Haenni, R.: 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)

[81] Spycher, O., Haenni, R., Dubuis, E.: Coercion-resistant hybrid voting systems. In: Krimmer, R., Grimm, R. (eds.) EVOTE'10, 4th International Workshop on Electronic Voting. pp. 269–282. No. P-167 in Lecture Notes in Informatics, Gesellschaft für Informatik E.V., Bregenz, Austria (2010)

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

[83] Spycher, O., Koenig, R.E., Haenni, R., Schläpfer, M.: Achieving meaningful efficiency in coercion-resistant, verifiable internet voting. In: Electronic Voting. pp. 113–125 (2012)

[84] Spycher, O., Volkamer, M., Koenig, R.: Transparency and technical measures to establish trust in norwegian internet voting. In: Proceedings of the Third International Conference on E-Voting and Identity. pp. 19–35. VoteID'11, Springer-Verlag, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-32747-6_2

[85] Tsiounis, Y., Yung, M.: On the security of elgamal based encryption. In: PKC'98, LNCS 1431. pp. 117–134. Springer-Verlag (1998)

[86] Volkamer, M., Spycher, O., Dubuis, E.: Measures to establish trust in internet voting. In: ICEGOV. pp. 1–10 (2011)

[87] Vollan, K.: Observing Electronic Voting. Nordem report, University of Oslo, Norwegian Centre for Human Rights (2005), http://books.google.ch/books?id=7jFqMwEACAAJ

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

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

Curriculum Vitae

1978	Born on December 30^{th} in Bern, Switzerland
1998	Matura Type C from Gymnasium Bern-Neufeld
1999−2002	IT-related position in private sector
2007	MSc in Computer Science from University of Bern
2008−2009	IT-related position in private sector
2009−2012	Research assistant at University of Fribourg and Bern University of Applied Sciences
2009−2015	PhD studies at University of Fribourg, Switzerland
since 2011	Position at the Swiss Federal Chancellery

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:
Trustworthy Internet Voting - Defeating Powerful Coercers and Vote-Buyers

Vorname:
Oliver

Name:
Spycher

Ort und Datum:
Fribourg, 31. August 2015

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".