University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Verifiability in the Swiss Remote Postal Voting System

Ivo Indergand
Zürich, Switzerland
Student ID: 12-923-892

Supervisor: Christian Killer, Bruno Rodrigues
Date of Submission: April 9, 2021

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Zusammenfassung

Traditionelle Wahlsysteme sind seit Jahren einem Vergleich mit e-Voting Systemen ausgesetzt. Aktiv werden e-Voting Systeme aber nur in wenigen Teilen der Welt eingesetzt, da vorgeschlagene Systeme oft fehleranfällig sind oder die nötigen Sicherheitsanforderungen nicht erfüllen. Das Schweizer Wahlsystem ist sehr komplex und unübersichtlich für den Wähler und basiert stark auf dem Vertrauen des Wählers gegenüber Drittparteien. Diese Arbeit versucht das gegenwärtige Wahlsystem verifizierbarer zu gestalten, indem sie kleine Komponenten des physischen Wahlmaterials anpasst. Konkret heisst das, dass zusätzliche QR Codes auf den Stimmrechtsausweis gedruckt werden. Diese beinhalten eine verschlüsselten Stimmregistereintrag des jeweiligen Wählers. Die entsprechenden verschlüsselten Einträge werden per IPFS auf einer authorisierten Website zugänglich gemacht. Diese Einträge kann der Wähler mit dem erhaltenen QR Code abgleichen. Darüber hinaus enthält jeder Stimmzettel einen RFID Tag. Mit Hilfe dieses Tags kann der Wahlvorsteher den korrekten Weg des Stimmzettels nachvollziehen, ohne dass die Identität des Wählers preisgegeben wird. Zusammengefasst wird nichts an der Gewichtung zwischen dem Stimmgeheimnis und der Verifizierbarkeit geändert. Der Fokus liegt darauf, das gegenwärtige System verifizierbarer zu gestalten, ohne jedoch einen Abstimmungsbeleg zu erzeugen. Da der Einsatz von RFID Technologie zusätzliche Kosten verursacht, bleibt es abzuwarten, wie praktikabel das vorgeschlagene Abstimmungsschema ist.

Abstract

Since many years, traditional voting schemes have been challenged by e-voting systems. However, e-voting systems are only actively used in some areas of the world, since proposed systems are often error-prone or do not meet the necessary security requirements. The Swiss remote postal voting is very complex (for the voter) and is heavily based on the voter's trust in third parties. This work tries to make the current voting system more verifiable by adapting small components of the physical voting material. Specifically, this means that additional QR codes are printed on the voting card. These QR codes contain an encrypted entry of the electoral register concerning the respective voter. The corresponding encrypted entries are made accessible through IPFS on an authorized website and can be compared with the received QR code. In addition, every PB contains an RFID tag. With the help of this tag, the electoral officer can reproduce the right path of the paper ballot without revealing the identity of the voter. Overall, nothing is changed in the weighting between privacy and verifiability. The purpose of the suggested scheme is to be more verifiable while staying receipt-free. However, since the use of RFID technology causes additional costs, it remains an open question how applicable the proposed scheme is.

Acknowledgments

I would like to thank my supervisors Christian Killer, Bruno Rodrigues and Prof. Dr. Burkhard Stiller from the Communication Systems Group at the University of Zurich. Especially Christian Killer provided me with a lot of support and gave an ideal guidance through the whole thesis.

Contents

Chapter 1

Introduction

Since the invention of democracy in ancient Greece, people have been able to cast their vote (although it was often restricted to a part of the whole society). Around 2000 years later, the emergence of the internet brought novel ideas and channels on how to conduct electoral processes. Remote Electronic Voting (REV) systems and schemes emerged, leveraging new cryptographic protocols achieving higher levels of privacy, verifiability and security. However, the increased complexities of such systems are still at the forefront of research and have cause political debates about voting software transparency [58]. The previous president of the United States of America (US) expressed repeated doubts about the expansion of Remote Postal Voting (RPV), claiming that there is enormous fraud involved [72]. In the US, facts indicate that there is fraud, but it does not seem to be a widespread problem [12]. However, the organizational overhead and the lack of verifiability are inherent to RPV and require trust in the processes. In the US, either absentee or mail-in ballots are used, depending on the state and jurisdiction. Indeed, using snail-mail to deliver ballots involves clear risks and trade-offs that need to be considered: it is slow and usually not verifiable (whether the ballot was received, counted and casted correctly) and there are inherent risks regarding delivery, storage and tallying process [10].

In general, RPV includes the distribution of paper ballots by postal services, which can be returned by postal service, or delivered in person to the electoral commune concerned [56]. In Switzerland, due to the federal and decentralized structure the cantons and municipalities are authorized to manage their respective jurisdictional electoral procedures autonomously [56]. For these processes, the cantons use centralized information systems [57], which can be used to transfer crucial data, e.g., in the case of elections, individual ballots are scanned or entered manually by a poll-worker and further evaluated on the centralized system. Also, intermediate results are transmitted through these systems [56].

There are different approaches towards making electoral processes more verifiable. These approaches often focus on specific systems in certain regions of the world, e.g., focusing on verifiability in voting booth systems. Other approaches improve on RPV systems by using cryptographic methods in order to increase verifiability and detect fraud and other error [50, 3]. Moreover, combined solutions, which use the best properties of the electronic and the snail-mail channel [39], were proposed. Another approach is called Code Voting. Thereby, voters receive a mail with an attached code sheet and communicate their choices

to the electoral authorities by inserting a code/with a code. This can be done by checking the returned codes correctness or via an untrusted electronic device [97]. Moreover, the in-booth elections can be improved. For instance, in the Wombat Voting System voters first identify themselves with their ID, then make a selection via the wombat's graphical user interface (GUI), and finally receive a foldable ticket including a QR (quick response) code and their selected choice. The QR code will further be scanned at the polling station, whereupon the selection will be separated and put into a cast vote box. At home the QR code can be verified in the browser [17]. Another attempt has been taken by Microsoft. In September 2019, Microsoft announced an open-source SDK called ElectionGuard, which is part of their Defending Democracy Program. The aim of this program is to make voting more secure, more accessible and more efficient. Moreover, it enables end-to-end verification of elections, passes results to third-party organizations for secure validation and assures voters that their votes were correctly counted [49]. However, ElectionGuard does not have the purpose/intention to replace paper ballots but rather supplement and improve systems relying on them [83].

The current Swiss voting landscape has mainly focused on preserving ballot privacy/receipt-freeness at the expense of verifiability properties. Privacy means that it is impossible to connect the voters identity with the filling of their votes [31]. It became mandatory for public elections, such as the secret balloting in Australia during the 19^{th} century, to prevent bribery and coercion [78]. Election verifiability is seen as a trade-off towards privacy and a potential balance shift raises questions about potential consequences. This adjustment between correctness and privacy has been done by most of the proposed voting schemes, e.g. [28]. Former work as in [29] has shown that a perfectly private audit trail (PPAT) is feasible and that an audit trail, be it electronic, paper or both, can realize universally verifiable elections. In their work [29] proposed a new encryption primitive which enabled to build the first universally verifiable voting scheme with PPAT while guaranteeing everlasting privacy. Moreover, they proposed two different constructions where one is tailored for small elections with homomorphic tallying and one for elections with complex ballots with mixnet-based tallying. These schemes achieve a workload for tallying authorities which grows linearly with the number of voters and candidates as well as a computational load not depending on the number of voters nor on the number of authorities [29]. Other voting protocols offering a PPAT required specific communication channels or a higher amount of work besides the voters which grow linearly with the number of trustees [29]. One option to offer blind signatures are PPAT where voters publish their ballots through an anonymous channel which takes care of the voting privacy, while the audit trail only contains anonymous information [29]. Setting up an anonymous channel for large scale election is, however, difficult. Another option uses a verifiable secret sharing scheme for the voter to distribute the information needed to tally their vote [29]. Those shares are then distributed to the authorities either protected by encryption or through private channels [29]. The solution given by [29] is based on the third approach of e-voting, namely the tallying of threshold encrypted ballots.

1.1 Description of Work

The overarching goal of this thesis is to offer a more verifiable and trackable audit trail for the Swiss RPV. This thesis should help answering the question if a more verifiable audit trail can be achieved without or with making small compromises regarding ballot privacy. This goal is broken down into smaller parts representing more concrete attempts that will be achieved in this work:

Background and Related Work: In order to design and propose highly sophisticated processes for a verifiable RPV, the relevant background and related work is researched and documented in technical depth. The focus of this step includes an in-depth research of the state-of-the-art in verifiable postal voting schemes as well as related work based on cryptographic protocols. This step also includes a clear documentation of the compatible protocols and cryptographic primitives to be used, as well as a classification of the most relevant voting schemes and protocols.

Design and architecture: In this part the design of a modular system architecture is suggested on which the RPV scheme can be executed securely and efficiently. Necessary cryptographic tools needed for RPV and corresponding software libraries that implement them have to be determined. In detail, this includes the respective evaluation of suitable cryptographic methods and protocols.

Implementation: This section is about the implementation of a prototype that works as a proof-of-concept (PoC) of the designed RPV scheme and design. This POC is delivered documented and as an executable open-source code in the end.

Evaluation, Documentation and Report: The prototype is evaluated with respect to the privacy, scalability, usability and verifiability properties provided by the chosen cryptographic primitives. In the end, a report that documents the findings of the initial literature review, design decisions, system architecture, source code, evaluation approach and evaluation results, and most importantly, the conclusion about the above overarching goal is presented.

Chapter 2

Background

This chapter introduces some key concepts of this use case. First, corresponding blockchain topics will be mentioned followed by an introduction into the concepts of verifiability and privacy. The last part provides an overview over the swiss voting landscape.

2.1 Blockchain (BC)

Blockchain became a very popular topic over the past years mainly because of the rise in cryptocurrencies. The underlying concepts of this technology, which are important for this use case, are discussed in this section.

2.1.1 Cryptographic hashing

A mapping from the set of all finite strings of characters from a first alphabet to a string of characters from a second alphabet of fixed length is called a hash function abbreviated as h [31]. The value h(x) for any x is called the hash value or message digest [31].

2.1.2 Blockchain properties

The ideas behind blockchain go back to the early 90's, when the proposal of [43] introduced a method for secure timestamping of digital documents [66]. In their scheme the server signs the current document with the current time as well as a pointer or a link to the previous document. Then, certificates were issued containing the aforementioned information [14]. These pointers, linking pieces of data, are created with the before mentioned hash functions. Later, this concept was improved: instead of linking documents individually, they could be collected into blocks which were linked together in a chain. Recursively the documents are again linked together in a tree structure [14].

2.1.3 Blockchain-based e-voting protocols

Before the blockchain technology was invented, the electronic voting protocols depended on public bulletin boards (PBB) in order to store data like census, votes and cryptographic proofs. Those boards were implemented as relational databases and did not allow data integrity and transparency. Blockchain avoids their single point of failure as well as their security issues by representing a decentralized PBB. This technique, however, also introduces computational complexity and limited scalability. The e-voting protocols use blockchain to store votes and calculate the tally with the aid of a smart contract's code without providing ballot secrecy [6].

2.2 Verifiability

In the context of elections verifiability is not a simple binary concept. There is a broad variety in verification empowering various people to verify several things under different assumptions [9]. Verifiability is defined in the following ways:

2.2.1 Individual and Universal Verifiability

In general, vote verifiability means that votes must be verified independently by their voters that were inserted in the final tally and have to be counted correctly [31]. There are two main types of verifiability (according to [31]):

- **Individual Verifiability:** each eligible voter can confirm that his or her vote was really counted.
- **Universal Verifiability:** anyone is able to check that the official tally is really the sum of all votes.

2.2.2 End-to-End Verifiability

End-to-End systems, also called universally verifiable voting- or receipt-based systems, do not derive their security from any specific type of voting equipment. Usually, they produce an encrypted representation of ballot choices that works as a receipt [23]. As stated in [12], an election with end-to-end verifiability accomplishes software independence together with the analogous notion of hardware independence as well as independence from actions of election personnel and vendors. Voting systems are end-to-end verifiable if they contain the following three types of verifiability (according to [59]):

- **Cast as intended:** Voters can independently verify that their selections are correctly recorded.

- **Collected as cast:** Voters can independently verify that the representation of their vote is correctly collected in the tally.
- **Tallied as collected:** Everyone can verify that any well-formed, collected vote is correctly part of the tally.

2.2.3 Hybrid Public Verifiability

Hybrid public verifiability of voting has been mentioned first under an approach called Proverum [57]. It is an attempt to address the lack of public verifiability in voting processes [57]. A system is offering Hybrid PV, when any person can publicly verify the accuracy of all administrative procedures executed [57]. The approach combines a private environment based on private permissioned Distributed Ledgers with a public environment stand on public blockchains and apply it to the Swiss Postal Voting system [57]. Therefore, Proverum allows the public to verify data within a public environment, while maintaining a privacy-preserving, verifiable audittrail within the private environment [57]. The final prototype models several Swiss municipalities implementing a private environment [57].

Hybrid Public Verifiability combines Administrative Verifiability (AV) and Public Verifiability (PV). The first one authorizes elections officials protection against errors and fraud and the second one allows that any individual can verify the accuracy of a tally [9]. Same as PBBs do in electronic voting, the Proverum architecture implements an immutable audit trail with multiple permissioned DLs for a clear distinction of a private environment and public BCs [57]. Moreover, Proverum provides trust, integrity, transparency and an architecture for a decentralized IdentityManagement [57].

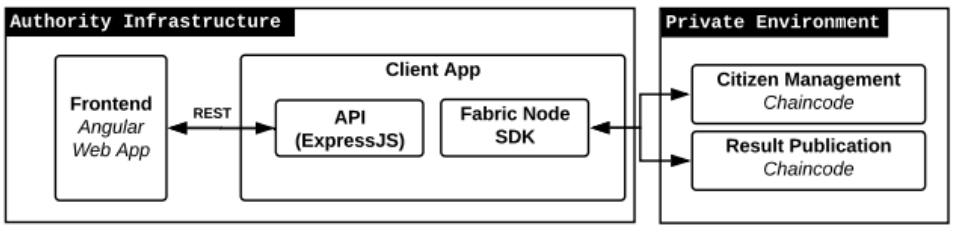

Figure 2.1: Overview Proverum Prototype (illustration adapted from [57])

2.3 Privacy

Privacy has been further separated into ballot secrecy, receipt-freeness, everlasting privacy as well as coercion-resistance [4]. To generate more trust, supporting technologies are used, *e.g.,* security tokens, smart-cards or paper-sheets containing codes like QR codes [79]. For example, based on Benaloh's Cryptosystem, an extension of the Goldwasser-Micali cryptosystem with the advantage that longer blocks of data can be encrypted at once, Microsoft has developed ElectionGuard [94]. This encryption enables the proof that a vote was not manipulated [89]. ElectionGuard is an open-source Software Development

Kit (SDK) enabling developers to build their own prototypes and was developed for the US poll site voting systems. ElectionGuard makes it possible to prevent systems being hacked, to alter votes by encrypting them, produces a paper ballot to deposit and its confirmation as well as a tracking code. The tracking code can then be entered online to check whether their vote was counted correctly or not [41].

ElectionGuard is using partially homomorphic encryption (HE) allowing direct operation on encrypted data [93]. HE comes in two flavours: partially HE and FHE [96]. Compared to the partial HE used by ElectionGuard, where one single operation can be performed on cipher text, IBM developed a toolkit to apply FHE that allows computing on encrypted data without decrypting it [46]. Therefore, third party service providers are able to perform certain type of operations on encrypted user data while the user's privacy is still guaranteed [95]. However, the performance of FHE is still quite inefficient [8]. Addressing that, HE is an act between utility, protection and performance at the moment [96].

Besides HE there are also other methods, like blind signatures, to reach privacy. Some of them will be discussed in the related work section.

2.4 Voting landscape Switzerland

The political system in Switzerland consists of three levels: government, cantons and municipalities. In the so called direct democracy, the swiss people are the major political instance of the country and can elect representatives of the people and therefore the swiss voter is able to vote on each level. On a national level, parliaments are elected through proportional representation and governments through majority representation [42]. The proportional representation distributes the statutory seats in the ratio to all cast votes to the different parties and elects 200 national councils representing cantons and parties. In the majority representation, the majority of votes, (*i.e.,* getting more than half of the votes), wins and elects the 46 council of states for example (excluding the cantons of Neuenburg and Jura). The majority representation is also used for electing cantonal governments and municipal councils [42]. The national councils and the council of states build together the federal assembly which is electing the federal council. Compared to that, the swiss people can vote the executive on their own at the cantonal level. There, the cantonal parliament represents the legislative of the canton and the cantonal government the executive of the canton [42]. Furthermore, to consider elections results more appropriate, some cantons have introduced the electoral process of the so called 'doppelter Pukelsheim'. According to this process, constituencies are gathered together for the counting of the seats which should cause a more proportional distribution of the elected parties and that smaller parties have better chances to gain seats. Compared to the cantonal level the elections on a communal level are highly diverse among the different cantons. Mostly the elections are either hold in a proportional representation or in a community meeting [42].

In list voting the name of a party has to be noted on a paper ballot when it is not already there. Without noting the party's name on the paper ballot, empty- or crossed votes get lost. In the voting envelope there are filled out paper ballots containing a number of a

list as well as an empty list paper ballot. There are 6 ways of voting. Let's assume that there are 4 national council seats, and one can vote for a list of a certain party. There are the following possibilities (according to [42]):

- Unchanged list: Without changing anything, the party obtains four list votes and each candidate of the party a candidates vote.
- Crossing candidates without replacing them: The crossed candidates are not getting a vote, but the party still receives the list votes for them.
- Crossing and replacing candidates: In the so called cross-voting a candidate becomes crossed and replaced from a candidate from another party. The party of the preprinted list loses therefore a vote and the newly placed party wins one.
- Accumulate a candidates vote: To support a candidate even more, one can cross another candidate and replace it with another candidates name already on the list. The party still realizes the same votes.
- Empty list with party mentioning: One can add an already existing list and party with individual candidates. Leaving voting slots empty these get accounted for the party nevertheless unless the whole list is empty. Accordingly, the vote is invalid.
- Empty list without mentioning a party: Without mentioning a party empty slots won't be accounted. This means that every party only receives the votes for the listed candidates.

In the end only one list can be handed in. According to the before mentioned facts it is evident that elections are more complicated in counting than voting with only yes or no possibilities.

Chapter 3

Related Work

This chapter introduces an in-depth understanding of cryptographic primitives, such as (Non-Interactive-) Zero Knowledge Proof Systems, Public Key Cryptosystems and Cryptographic Signature Schemes and how they can be applied to the Swiss RPV. The section also includes the exploration of additional methods to achieve Hybrid Public Verifiability, as well as End-to-End Verifiability in RPV [57], where the limits of those reside, and whether they could be integrated with the Swiss RPV itself. Moreover, various voting schemes are compared and evaluated, with a focus on their application in Remote Postal Voting.

3.1 Comparison and evaluation of voting schemes

There has been a lot of activity in the field of verifiable voting schemes over the past decades [76]. Generally, voting schemes can be classified due to where the voting is happening and how the ballot is cast, either on a physical or a virtual ballot. Therefore, [52] suggested four categories of voting schemes: in-person paper voting, in-person e-voting, remote paper voting as well as remote e-voting.

An electronic voting system is a popular application of cryptographic tools which are studied by several researchers [48]. Many protocols have been proposed over the years [48]. The proposed protocols use a variety of cryptographic techniques [24]. These cryptographic protocols represent one part of a larger system consisting of voting machines, software implementations and election procedures [53]. Their security must therefore be analyzed on its own but also by taking the whole system into account [54]. For security purposes, mainly three election models are used: the mix-net-, the blind signature- and the homomorphic encryption model as well as eligibility tokens [67]. From those, only homomorphic encryption (HE) supports direct tallying without decrypting every vote [67]. HE enables computation on encrypted data even on clouds, producing an encrypted result which can then be decrypted [41]. Due to the fact that HE makes voting tabulation straightforward HE represents a perfect fit for election security. It is a simple form of HE. Allowing addition and multiplication capabilities at the same time makes full HE

(FHE) the Holy Grail of cryptography and cloud security [41]. Craig Centry made the combination in FHE possible in 2009 [38].

The comparison of different voting schemes concerning cryptographic properties are summarized in the tables 3.1 and 3.2.

	Primitives used					
Voting Schemes	Zero Knowledge Proof	Blind Signature Scheme	Homomorphic Encryption	Mix-Net Scheme	RSA Signature Scheme	El Gamal Cryptosystem
Foos Scheme [36]	x	x				
Radwin Scheme [71]	x	x			x	
Juang and Lei's Scheme [51]		x			x	x
Cramer *et al.* Scheme [28]	x		x			x
Prêt-à-Voter [77]	x			x		

Table 3.1: Schemes and primitives (Illustration adapted from [30])

The definition of those security properties according to [31] are as follows:

- Eligibility: Means to fulfill the requirements of being eligible to vote.
- Fairness: No participant is able to gain any knowledge about the tally before the counting stage except for his own vote.
- Verifiability: See the definition in subsection 2.2.1.
- Voter-Privacy: It must be impossible to connect the content of his/her cast vote to the voters identity while it must be ensured that the voter can cast a ballot.
- Receipt-Freeness: A receipt cannot be constructed by a voter to prove his/her vote to a third party. This prevents vote selling or buying.
- Coercion-Resistance: This ensures that an elector cannot work with a coercer to show that he/she voted in a specific way.

	Cryptographic properties						
Voting Schemes	Eligibility	Privacy	Individual Verifia-bility	Universal Verifia-bility	Fairness	Receipt Freeness	Correct -ness
Foos Scheme [36]	x	x	x		x		
Radwin Scheme [71]	x	x	x				
Juang and Lei's Scheme [51]	x	x	x				
Cramer *et al.* Scheme [28]	x	x		x			
Prêt-à-Voter [77]		x	x	x		x	

Table 3.2: Cryptographic properties (Illustration adapted from [30])

3.2 Verifiability versus Privacy/Receipt-Freeness

Achieving a good balance between verifiability and receipt-freeness has opened - and still opens up ongoing questions to a lot of researchers. [80] identified three non-exclusive solution categories for blind signature e-voting protocols in which research has been invested. First receipt-freeness, meaning the avoidance of creating any form of receipt, using mostly homomorphic encryption. Here, as history has demonstrated, the assumption of totally safe channels for data transmission is an issue in practice. Assumed save channels were not as save as expected. Moreover, a collaboration of the signing and voting authority would lead to the same result as a compromised transmission channel. Secondly, instead of working with no receipts one can avoid creating atomic identifiers so that no link is identified between ballot and voter (like its done with separate ballots in Switzerland, for example). The third option was constituted by [80] to make receipts hard to abuse. [20] for example developed a voter-verifiable scheme which hides the receipt information in several pieces, only offering information when all its parts are combined. The missing blinding factor, however, leads to severe disadvantages and requires full trust in the authorities [80] .

3.3 Theorie and Technologies

The following section contains several topics concerning theorie and technologies for this use case. The subsections mostly build upon topics from previous subsections.

3.3.1 Public Key Infrastructure (PKI) and alternatives

A public key infrastructure can be interpreted as a store of different internet technologies which provide secure network communications. The four key elements of a PKI are digital certificates, public and private keys, certificate authorities and certificate revocation lists [1]. A blockchain network contains different actors like peers, orderers, client applications, administrators and more [1]. All of these actors have digital entities covered in X.509 digital certificates. Therefore, a PKI offers a list of identities which regulate the permissions over information access and resources [1]. Regarding voting a PKI should satisfy two properties: efficiency and reliability. Efficiency means that the used voting protocols should gain the information needed as fast possible from the PKI. Reliability states that corruption in the components of the PKI should not expose the voting process to risk [30].

The PKI uses a digital signature technology, meaning public key cryptography. The main idea of PKI is that the secret private key of each entity is only known by that entity [82]. The derived key of the PKI, the public key, can not be used for signing but verifying signatures. Therefore, the public key is visible to anyone and normally included in the certificate document. In this message exchange the digital certificate is used to authenticate themselves. The certification revocation list composes a reference for certificates which are no longer valid [1].

Compared to the three security procedures of public key encryption (key generation, encryption and decryption), the HE scheme contains four procedures [84]. PKI alternatives, such as 1Password, often use multi-factor authentication (MFA). MFA requires additional authentication measures for access to sensitive information. Instead of only using username and password users can be prompted to provide SMS codes, biometric informations or email confirmation actions to verify their identities [37].

3.3.2 Pseudo-random functions (PRF)

A function family is a map F equal to T x D -> R [81]. In this equation T is the set of keys, D the domain and R the range [81]. A keyed PRF is used later in this work where an HMAC function is composed by a hash function which is parameterized by the symmetric key k [81]. A PRF is an efficiently computable function where a random instance of its family is computationally indistinguishable from a random function while the key remains secret [81]. Therefore, a hashed message authentication code (HMAC) is a PRF whose resistance against collision is one of the underlying hash scheme [81].

3.3.3 Zero Knowledge Proofs (ZKP)

ZKP are often applied in e-voting protocols [26]. ZKP are used to uncover and proof information without revealing the underlying content. There are two kinds of ZKP: an interactive and a non-interactive type. Both have two parties involved, a prover and a verifier. The interactive ZKP two-party protocol consist of three moves called the commitment, challenge and response. On the other hand, in the non-interactive ZKP variant the prover generates the proof once and sends it to the verifier in one round [60]. Therefore, the proof can be reused, compared to the interactive variant in which the proof has to be repeated. To move from an interactive proof to a non-interactive one the Fiat-Shamir heuristic is applied regularly where the inputs of the prover are hashed to simulate the randomness of the verifier [26].

3.3.4 Signature schemes and anonymous channels

Digital signatures enable petition security and the protection of data integrity and serve as one of the most substantial applications of cryptographic protocols. Blind signature schemes can be used for an election authority to certify the ballot and to verify the voters identity [90]. They often make use of public key signing -and cryptographic protocols [90]. A digital signature scheme contains three algorithms: a key generation-, a signature- and a verification algorithm [11]. It is considered secure if an attacker is not able to produce a valid signature despite having access to a signature oracle [40].

David Chaum has introduced blind signatures and its application in the payment system in 1981 [18]. The authors aim was that the bank is able to sign a cheque for a mandated payee to a third party without letting them know who the payee is while still proofing the payees payment. Therefore, the blind signature method preserves the privacy and anonymity properties of [80].

Generally, all signature schemes are building upon public key cryptography which was introduced in the subsection before. The first one was implemented based on a RSA scheme, with the only difference of containing a blindness random value [19]. Blind sign a message takes six steps for calculation. There are three actors in this scenario: an author is blinding the intended message, a signer of this message and a third party that verifies the signature. The privacy of this blind signatures can be hidden, weak or strong [80]. Regarding our remote postal voting setting, the blind signatures are used to guarantee the used protocol's privacy.

The first time blind signatures were proposed for e-voting protocols was in 1992 by the so called Foo paper [80]. In [36], the authors only introduced the crypthographic methods by using a generic protocol, without defining a concrete one. The simple scheme combined techniques of blind signatures and anonymous channels which became the basis for other following-up implementations. Since then the two major developments were made in anonymous channels/blinded signatures and homomorphic encyrption functions [75]. The blind signature e-voting protocols suggested by [36] can be seen in figure 3.1.

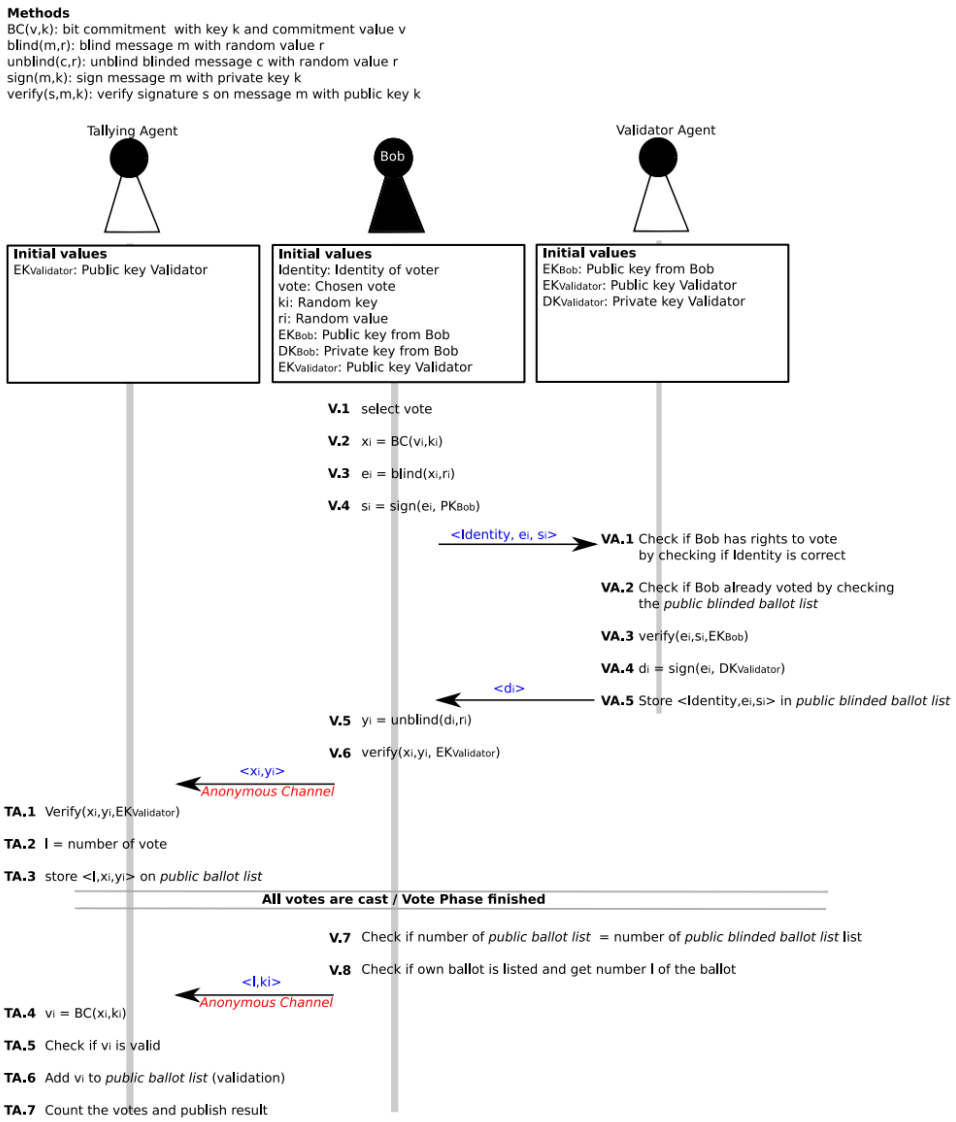

Figure 3.1: Blind signature e-voting protocols suggested by Foo (illustration adapted from [80])

By implementing blockchain-based e-voting protocols and deploying the blockchain as the bulletin board one eases security aspects as data integrity, transparency or fault-tolerance [6]. Trials have been done by [65] by implementing the OVN or under the aegis of [5] implementing an ECC blind signature scheme on Ethereum where the last one satisified the secret ballot, verifiability, practical and fair criteria for using a protocol [6].

Using blind signatures make the attempt to achieve receipt-freeness impossible. However, when removing those signatures, one has to find other techniques to verify the scheme. Future developments seem to offer more efficient zero-knowledge proofs and encryption techniques and will struggle to find a verifiability level allowing for receipt-freeness [75].

Anonymous channels are used to anonymize a message sent between peers, in order to make it impossible for the root to be traced back [80]. One of the most infamous alternatives and entry to the 'Darknet' is The Onion Routing Protocol (TOR). TOR is a

circuit based low-latency anonymous communication service where the 'onion routing' is the main principle [32]. Another alternative is the bit commitment scheme to secure the communication between the voter and the authorities servers [30]. A bit commitment is a scheme where someone makes a commitment and conceals the commitment from the public until the person decides to open it [80].

3.3.5 Interplanetary File System (IPFS)

IPFS is a peer-to-peer (P2P) hypermedia distribution protocol to make the web faster and safer. Moreover, IPFS serves as a distributed system to store and access files, websites, applications and data [47]. Compared to the world wide webs structure on ownership and access, IPFS is based on the ideas of possession and participation [47]. This means that many people possess each other's files and participate in making them ready for the use of others [47]. Instead of identifying a file's location, IPFS addresses a file by its content. This content identifier is a cryptographic hash of the contents address [47]. Overall, one could consider IPFS as a decentralized storage offering to store webpages as a mirror.

3.4 Data model

The eCH union regulates e-Government standards to attain an effective electronic collaboration between authorities, companies and privates [33]. The crucial standards for this use case are eCH-0110 [35] and eCH-0220 [34].

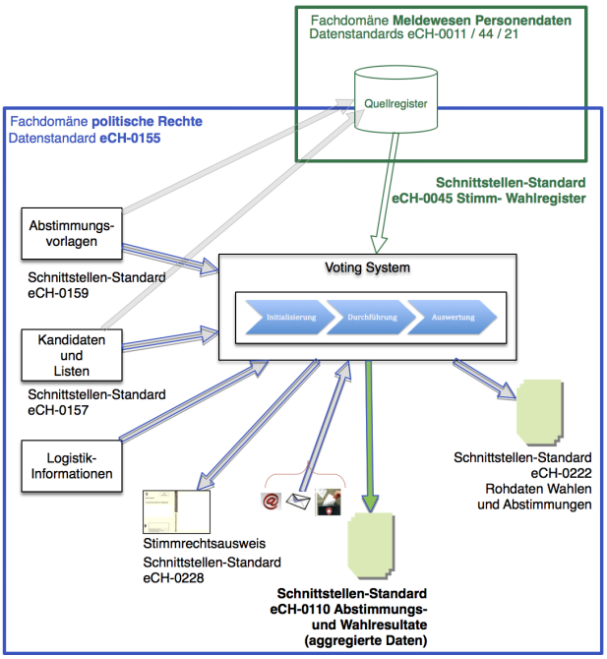

Figure 3.2: Swiss e-Government Data Standard for Political Rights (illustration adapted from [35])

[34] represents an instruction guide for the conservation of the validity for electronically signed documents. Since the relevant documents are stored on IPFS and referenced by the public ethereum blockchain, information about each document can be accessed through the underlying contracts and their transactions.

Chapter 4

Use Case Analysis and Requirements

This chapter describes the current situation of the Swiss RPV and a possible extension using either crypographic means or by not using them. Moreover, the requirements are covered, and a final analysis is done which serves as a basis for the design proposal.

4.1 Swiss Remote Postal Voting

As the Proverum approach states, there are interesting possibilities on how the Swiss federalism can rebuild their vote infrastructure on a blockchain basis [57]. Compared to Proverum, which focused on implementing the private voting environment, this thesis focuses on finding alternatives for the public environment of the voting process.

4.1.1 Current Swiss Remote Postal Voting

The Swiss RPV is simply paper-based and the end-to-end Postal Voting Process Flow (PVPF) is split into six main phases with corresponding sub-stages [56]. This is summarized in figure 4.1. As illustrated, there are many security threats involved in the different phases which are either inherent to the system or due to external suppliers. This supply chain therefore demands trust between those stakeholders and no malicious intentions. The stakeholders involved in the Swiss RPV are displayed in figure 4.2. The external service providers (ESP) can be distinguished into artifact manufacturers, the Swiss Post and software providers for E-Counting (EC) tools and communication devices.

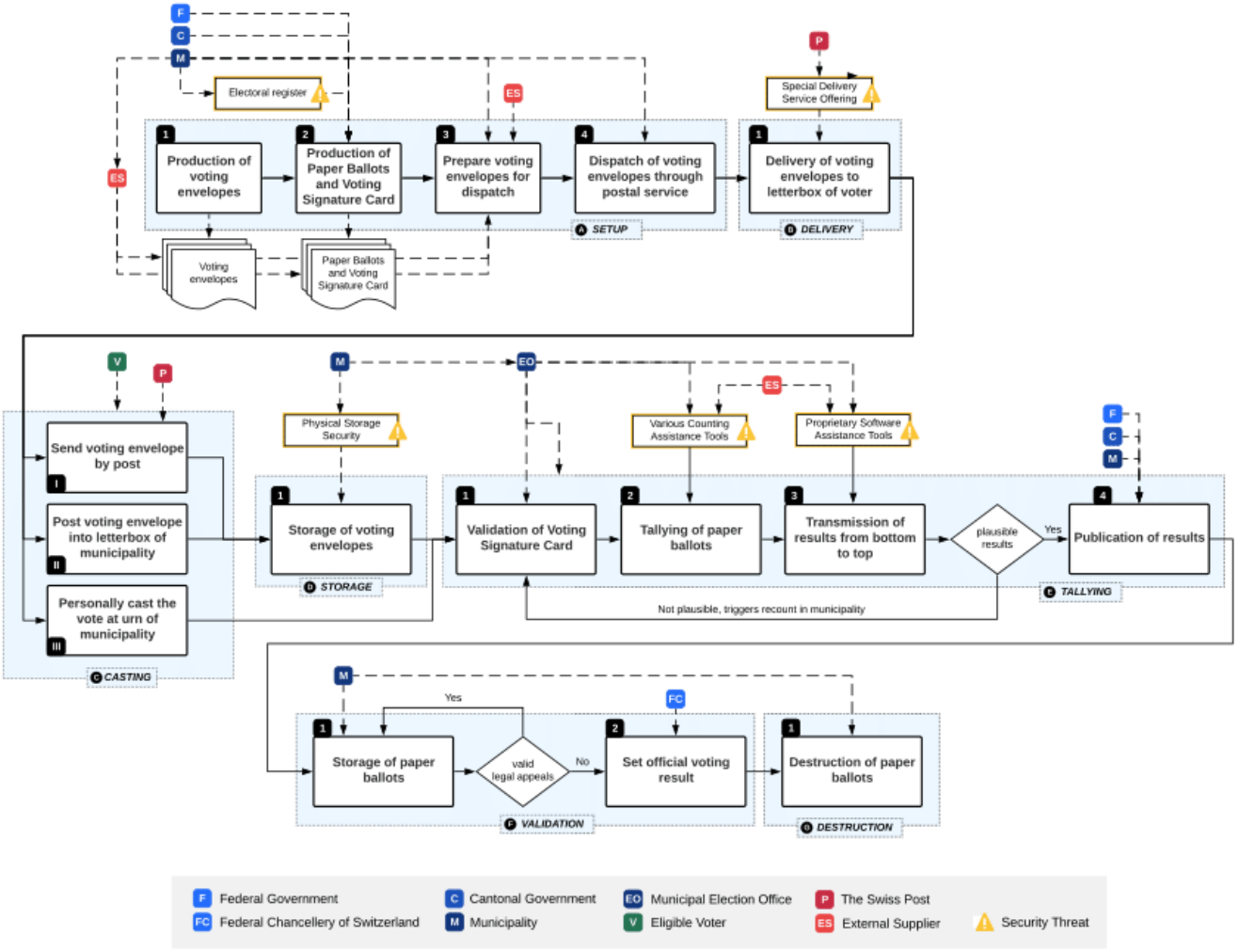

Figure 4.1: Swiss RPV Process-Flow (illustration adapted from [56])

Additionally, several voting schemes have already been compared on 13 criteria in the suggested framework of [52] as shown in figure 4.3. There, the author claims that the Swiss RPV is highly insecure due to issues concerning vote-buying, coercion and the dependence of the integrity of the results on trusted authorities. In their verifiability properties a voter can not ensure their ballot is not accidentally spoiled and can not ensure that their vote is recorded as cast. Moreover, considered the authority is not corrupted, the author supposes that a voter can detect if their vote is displaced, the tally is counted correctly from the recorded votes and no ballot stuffing is happening.

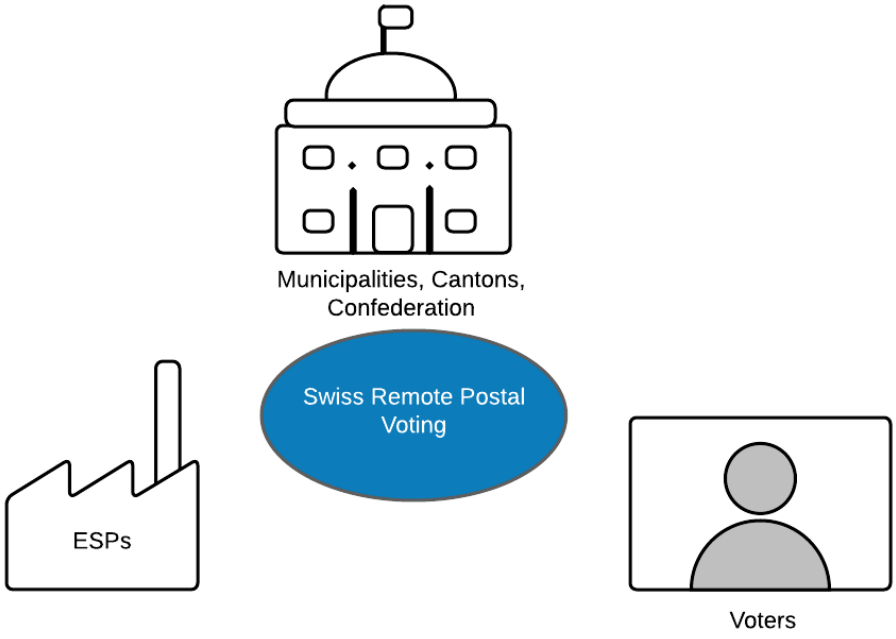

Figure 4.2: The stakeholders involved in the Swiss RPV

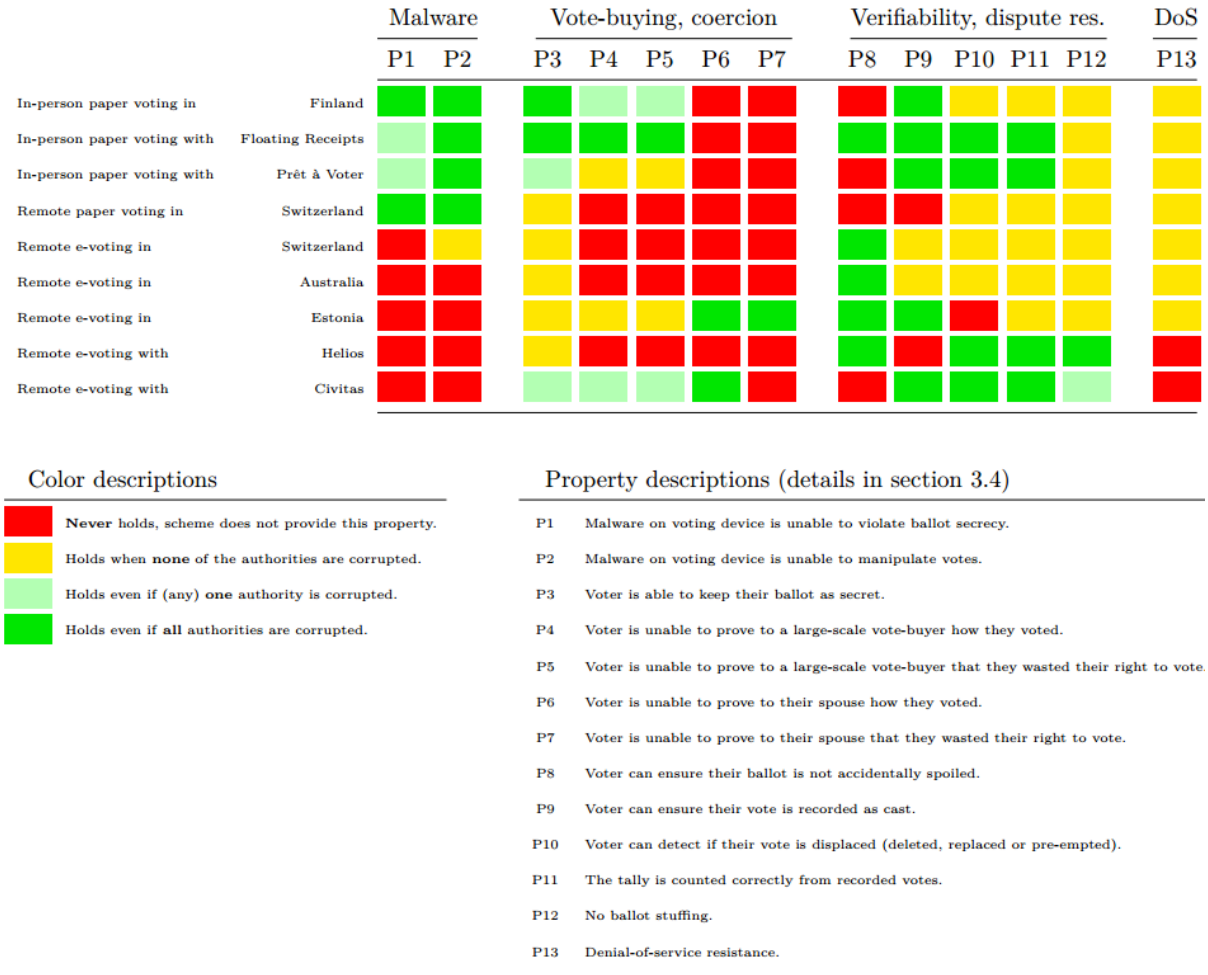

Figure 4.3: Voting scheme comparison (figure adapted from [52])

4.1.1.1 E-Counting

Besides the already mentioned pilot projects in the e-voting area there exists also E-Counting. [69] stated in 2017 that over ten percent of the delivered paper ballots are no longer counted manually, but that they are scanned and electronically evaluated. However, the requirements of the government towards the EC of voices is insufficient and not advisable. Central aspects to approve the counting result, like the four-eyes principle or the requirement of a statistically relevant sample, are missing. Moreover, specific requirements for EC compared to a possible e-voting system are not given by the federal council, which seems insufficient. Also, the ESP's, providing the web-based assistance tools, are fulfilling the international Good Practice [69]. Practice has shown that electronic systems still have difficulties to correctly detect hand-marked ballots and that paper ballots should be tested before scanning. Although the Swiss cantons and municipalities are enforced to develop own solutions, it is difficult to push them for higher standards [69]. So far, the city of St. Gallen, Lucerne and Basel-Stadt as well as Bern have used EC. They implement/-ed EC with scanners, laptops and DB-servers in an offline state. Conventional methods before were using precision balances, bank counting machines or counting by hand [91]. The EC software applies either Intelligent Mark Recognition or Optical Marc Recognition which both detect markings in the form of caskets, crosses or barcodes [69].

4.1.2 Extending the Swiss Remote Postal Voting

Often used approaches to make paper-based voting systems more secure is cryptography in form of digital-pen-, punched card-, optical scan- or scratch-card voting systems [73, 21, 22, 25]. However, the use of this systems is limited to deployment, usability and security issues [2].

Concerning this use case, a new 7^{th} phase can be introduced between the Delivery and the Casting processing steps. There are several possibilities to extend the current voting process by either a physical, a cryptographical or a combined approach of both of them. Hereinafter, different ideas, on what a new phase could look like, are presented.

In Switzerland every eligible voter receives four different paper artifacts delivered by the Swiss Post: a two-way voting envelope (VE) containing a voting signature card (VSC), a paper ballot envelope (PBE) and paper ballots (PB's). Each VSC is printed from the informations it gains from the electoral register (ER) [56]. The artifacts shape can be seen in figure 4.4.

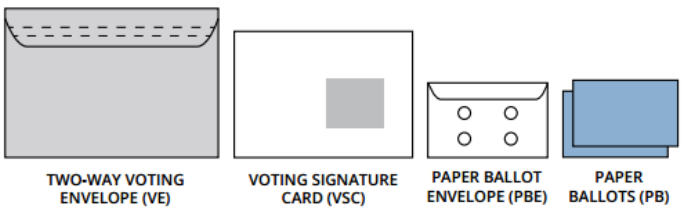

Figure 4.4: Paper artifacts (illustration adapted from [56])

Each municipality has an own template for their VSC's but all of them contain the name and the address of the eligible voters. It is the only artifact that is signed by the voter and guarantees the validity of the vote [56]. The PB's on the other hand are in the exact same manner for every voter. In this use case we do not intend to propose a new paper-based ballot voting design, but try to improve traceability and verifiability of the current system and make it more robust.

4.1.2.1 Possibilities with encryption

Generally, the most challenging part is to allow voters to check if their votes were cast as wanted. Previous work encouraged the voter to perform a randomized protocol for testing and verifying [10]. As mentioned in the previous chapters, Microsoft's ElectionGuard tries to enhance US elections verifiability by using QR codes in polling stations. Josh Benaloh, the provider of the conceptual and mathematical basis for end-to-end verifiable elections for ElectionGuard, was also involved in other approaches towards verifiable remote postal voting ([10],[16]). In his 2013 released proposal 'Verifiable Posting Voting' he and his co-authors tried to combine the best values of paperbased and end-to-end verifiable remote voting systems. Therefore, ballots would be delivered electronically to voters who are returning their votes on paper together with some cryptographic information for verification purposes [10]. Concerning the voter's experience, the voter only needs a simple check that the human-readable printout reflects the intended vote [10]. Accordingly, the additional work means to add some cryptographic information to the envelope containing the human-readable vote. One approach can be the delivery of ballot information online, letting the voters print and return paper votes by mail eliminating the first and more difficult half of the snail-mail delivery. This approach allows simple cast-as-intended verification without enabling more guarantees regarding privacy and so on. Another approach is to use cryptography to decrease e-voting vulnerabilities. Now, the authors try to combine both approaches using a simple cast-as-intended check for most voters and a verifiable protocol for verifiability through an electronic and a snail-mail channel [10].

As the EC shows, there could be another possibility of using the already existing equipment and add additional feature to the counted paper ballots. One could add a physical artifact or combine all paper ballots in one having machine readable caskets, crosses and barcodes. Such systems already exist in Swiss areas. This could be augmented by an additional QR code, which can be scanned like the rest of the paper ballot, to achieve universal verifiability. QR codes can contain digits like phone numbers or URL links. VotingWorks, an open source vote-by-mail system for e-voting, for example has been criticized because QR codes are unreadable to the eye and that it is therefore impossible to represent the voters' intent properly [44]. Since this prototype only uses the QR code for a double check in tallying, this concern should not be shared in this case. Like [85] showed in his blockchain adaptation of the current e-voting system in Estonia, one could generate a QR code from a randomized session code.

There are several scenarios on how to include cryptographic signatures in the current Swiss RPV process. Liu and Wang proposed a simple blockchain-based e-voting protocol in combination with blind signatures in [61]. They divided the voting process in three phases and distinguished between three actors, namely voters, organizers and inspectors.

Starting in the pre-voting phase eligible voters are authenticated and registered by an identity management system. In the main voting phase the organizers and inspectors sign the hashed vote based on the registration of the voter before and therefore the blind signature protocol is applied. Sending the vote from a newly generated address works essentially as a mix-net in the essence. In the counting phase, the votes with invalid blind signatures can be excluded. This paper/protocol has been implemented with the Etherum Virtual Machine (EVM), which is able of verifying RSA signatures, in a POC [7]. Before, there was an implementation of the Open Network protocol (OVN) by [65] which, however, was not gas-efficent (gas meaning the cost of performing a transaction on the network) [5].

An alternative physical variant could contain an RFID (radio-frequency identification) tag on the PB. The authors of [40] introduced two systems of tracking paper-based ballots: one with digital signatures and one using a Tracker-based system, based on the work of [13]. They follow up on the idea of treating the voting system as a logistics system. The tracer uses RFID tags that store a state which can be read and updated while the artefact moves. In this system, an initial tag state stores a polynomial evaluated at a value chosen by the issuer that is called the pathmark [40]. The readers' keys and arithmetic operations are used to update the pathmark when they receive a ballot. This describes where the goods have travelled [40]. Hence, the central idea is to encode valid paths with polynomials in supply chains [13]. Since the Tracker system uses HE, the system can not be applied on this rather simple use case. This use case applies NFC, a subset of RFID which is able to read and write, to make calculations on the tag while scanned by a reader device. The pathmark suggested by [40] would be calculated through another mechanism or a simpler algorithm. With Tracker, the RFID tag only needs to store the encrypted ID, the encrypted HMAC and the encrypted path mark [13]. Those three Elgamal ciphertexts would need a storage of 960 bits. Moreover, the complexity of readers would also be low and would need less than 80 bytes to store the Elgamal public key and part of the polynom [13]. Therefore, a large Tracker system with 10^9 different tags and 10^6 different valid paths would only consume around 11 gigabytes of storage on side on the manager/authority side [13]. RFID Track Ballots were already used in Costa Rica's elections for the ballot containing bags or in Alameda County, California. During the election process, an RFID reader can bounce a radio signal off an entire bag of assets at once and record the contents instantly in an asset tracking system [27].

4.1.2.2 Possibilities without cryptography

A more 'academic' than 'practical' approach towards new paper-based voting methods has been proposed by [74]. In those proposals, every voter can verify that their vote is recorded as intended and every voter gains a receipt to check later if their vote is included in the final tally as well [74]. This works without the voter being forced to show how he/she voted to anyone else. To make the election result calculable for everyone, the cast ballots are scanned and published in plaintext on a PBB [74]. In ThreeBallot, every voter directs three paper ballots. This underlies certain restrictions on how they should be filled out. The voter then copies one ballot for his/her receipt which, however, does not show how a person voted. Only the voter knows which ballot was copied [74]. If the receipt

does match the PBB the PBB has been counted the right way. Otherwise, deletion or modification of the ballots is detectable. Similarly, attempts like VAV and Twin exist [74].

4.2 Requirements

Certain functional (FR) and non-functional requirements (NFR) have to be fulfilled for this use case as listed in table 4.1 and table 4.2. The POC will gear to those requirements.

4.2.1 Functional Requirements

FR Nr.	**FR**	**Definition**
FR1	Storage	A storage for the uploaded documents is offered by the authorities.
FR2	Verifiable ER (VER)	A verifiable ER is created by the authorities.
FR3	Verifiable VSC Register (VVR)	A verifiable VSC Register is created by the authorities.
FR4	RFID tagged ballots	Ballots are supplied with included RFID tags from ESP's.
FR5	RFID authentication	Authentication must be satisfied for RFID tags.
FR6	RFID unlinkability	RFID tags must be unlinkable.
FR7	RFID privacy	RFID ballots must guarantee privacy.

Table 4.1: Functional Requirements

4.2.2 Non-Functional requirements

NFR Nr.	**NFR**	**Definition**
NFR1	Temporary VER	The VER will be deleted after the election and is lasting only a short while.
NFR2	Temporary VVR	The VVR will be deleted after the election and is lasting only a short while.

Table 4.2: Non-Functional Requirements

4.3 Analysis

The analysis has demonstrated that Switzerland focuses on achieving voter privacy at any cost. The mentioned research has shown that creating a more verifiable and trackable RPV is only possible to a certain degree (without creating any receipt). According to this fact, privacy should remain at the current status while gaining some level of higher verifiability and only make small changes, as the trade-off is hard to be shifted. Therefore, an electronic audit trail is added to the current physical audit trail in the Swiss RPV in form of a code and tracking ID's on the VSC as well as on each paper ballot. The ID does not reveal any information about the voter and its sole purpose is to a) create time stamps and b) compare the counting accuracy.

Chapter 5

Design proposal

Based on the performed use case analysis, it was identified that the current Swiss RPV could be augmented by either electronical or physical components or both of them. Accordingly, the suggested design is described and documented in this section.

5.1 RFID-based Ballot Tracking Proposal

A paper-based voting system can also be seen as a non-trivial, security-critical logistics system for transporting ballots from ballot boxes to election offices [40]. As mentioned in the use case analysis, [40] introduced cryptography to improve the strength of the logistics part of a paper-based voting system. To fit into our suggested scheme, we adapt and change their scheme accordingly. The following three main entities and stakeholders are mentioned in the tracking model (according to [40]):

- Election Issuer: The central party which prepares the ballots being deployed into the voting chain. Adapted to our scheme, these are the ESP's who are managing and distributing the key material.
- Election Officials: An intermediate person at a given place who interacts with the ballot in some capacity at this place. Those people are poll workers, for example.
- Election Manager: The election office manager receives the ballots at a certain checkpoint in the chain and checks the signature on each ballot in order to ensure their legality. By doing that, it is verified that each ballot has passed through a valid path in the chain until reaching that checkpoint.

The suggested Tracker system of [13] has been improved twice. [92] improved the former work by proposing a more efficient tag path authentication protocol by reducing the computational overhead and memory requirement on the tags and decreased the memory space from 960 to 800 bits. Then, [70] further reduced the memory space from 800 bits to 640 bits while also removing the strong assumption that the manager does not need

to store all readers' private keys [70]. The implementation of [70] will be discussed in the next subsection.

The tracker version of this use case will be called Ballot Tracker. Electronic product code (EPC) class one tags of the second generation will be used for the actual use case, as mentioned in [70]. They underlie the ISO 14443 norm for contactless chipcards like credit cards do [87]. To enable the new voting scheme, it is important to choose a trustworthy technology to develop a possible electronic ID and/or tracking system. Therefore, QR and RFID technologies have been compared regarding their capabilities and advantages and disadvantages, as detailed in tables 5.1 and 5.2. Due to the individual advantages and the usability for this use case, the VSC's are augmented by a QR code while the PB's are each labeled by an RFID tag.

RFID uses radio waves to transmit and identify objects and is supposed to replace barcodes soon. Normally, RFID involves an RFID tag, which consist of an antenna and a chip which is attached to the tracked object [68]. The chips store around 2kb of data and are open to cost and privacy issues. Therefore, light-weight authentication protocols are used to handle tag-level constraints. On the other hand, the tag readers are connected to a backend server and a database for the processing [68]. To avoid using a tag reader for every vote, counters and election officials two RFID tag gates can be installed at each of the 12 regional offices and bulk scan the PB's. One scans the PB's when they arrive at the municipality, and the other one does so after they have been counted.

Advantages	**Disadvantages**
Cheap labels	Intervisibility needed
Technology well-established	No pulk acquisition
More information than bar codes	Sensible to pollution
Easy production	
High spread of suitable readers	
Flexible use on arbitrary areas	

Table 5.1: Pros and cons of QR codes (according to [15])

Advantages	**Disadvantages**
No intervisibility needed	More complex and expensive than QR codes
Nearly 100 % recognition rate	Depending on RFID type sensible to metals or liquids
Fast data exchange	
Big distances possible between transponder and reader	
Not sensible against pollution, smaller damages and environment constraints	
Coverage is feasible in real-time	

Table 5.2: Pros and cons of RFID (according to [15])

In this use case there are three different authorities processes: one for the VSC's, one for the PB's and one for the tallying. These processes are summarized in the voting scheme in figure 5.1 and the involved stakeholders and actions are described in Table 5.3. Green lines indicate the election office, red lines the voter as a stakeholder in figure 5.1.

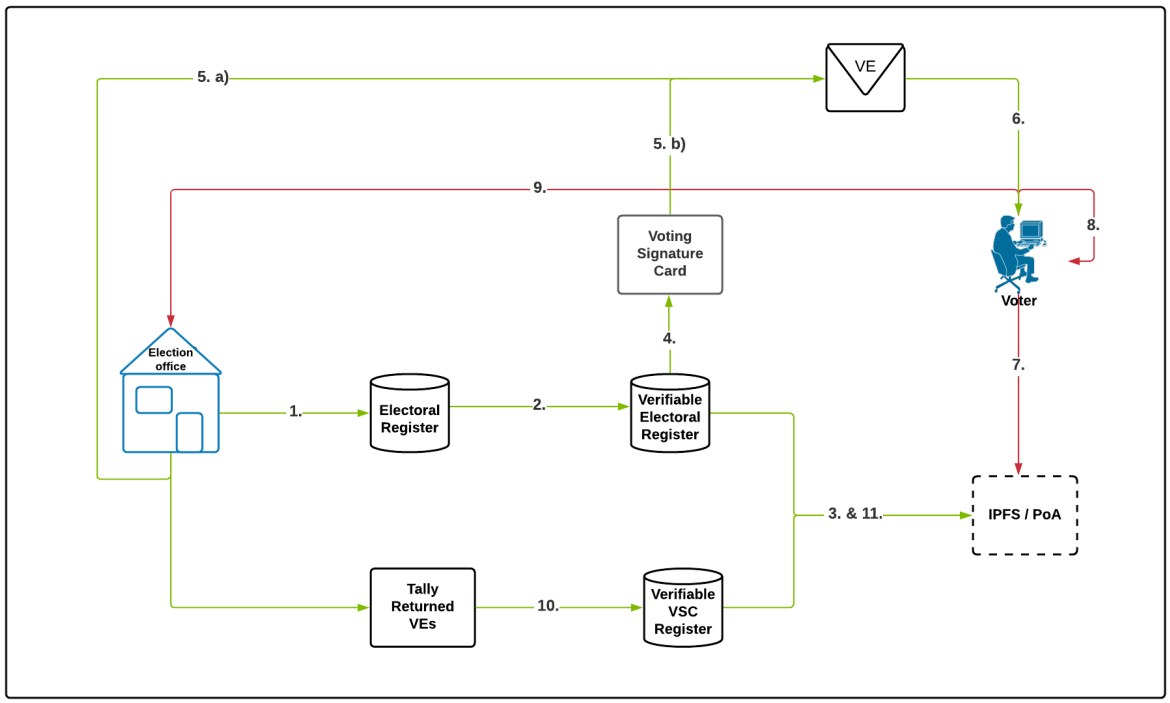

Figure 5.1: Voting scheme

In the authorities VSC process, the electoral register is created in the first step (1) (*cf.* Table 5.3). Since the residential municipality contains all the residential information in the residents' register, the electoral register can be exported. The residents register declares which resident is eligible to vote and has to receive a VE from the authorities. To protect the register from Brute-Force attacks, each entry is hashed and signed with a salt in step two (2) (*cf.* Table 5.3) before the VER is put on the VSC in form of a QR code in step four. Brute-Force attacks describe the event when one tries to find keys or passwords through trial and error. This hashed and signed list is then published to IPFS through the ethereum smart contract in step three (3) (*cf.* Table 5.3) and is called verifiable electoral register. That concludes the first process.

In the second process, the PB is physically sent from the election office together with the VSC in the VE in step five (5) (*cf.* Table 5.3) after the signature has been printed on the VSC in step four (4) (*cf.* Table 5.3). The PB got already tagged with an RFID tag, which gets scanned later at the municipality count optical scanner, as well as before the manual tallying at two RFID gates. After the publishing from the side of the election office, the voter receives the VSC, PBE as well as the PB's. First, the voter fetches the VER QR code from IPFS and scans the QR code on the VSC, whereupon he/she registers the signed hash of the authority in step seven (7) (*cf.* Table 5.3) and step eight (8) (*cf.* Table 5.3). For that purpose, the VER is downloaded and compared to the signature.

Step	Stakeholder	Action
1.	EO	Exports ER from citizen registry
2.	EO	Hash and sign individual entries
3.	EO	Publish VER to IPFS
4.	EO	Print signature on VSC
5.a)	EO	Include RFID tagged ballot into VE
5.b)	EO	Include VSC into VE
6.	EO	Send VE via postal mail
7.	Voter	Fetch VER data
8.	Voter	Scan and save signature
9.	Voter	Return VE via postal mail
10.	EO	Scan and sign tally returned VE's
11.	EO	Publish VVR to IPFS

Table 5.3: Voting scheme setup

According to that, the voter now knows that the VSC is valid and needs to store the value checking if the VSC has arrived at the election office later. This completes the second process.

The authorities' tallying process as the closing process, receives the whole VE from the voter in step nine (9) (*cf.* Table 5.3). The tallying itself is heterogeneous among cantons and municipalities and is not regulated on a federal level [56]. Therefore, the election office defines the details of the tallying process [56]. Before publishing it to IPFS in step eleven (11) (*cf.* Table 5.3), the signatures will be stored to an offline storage (like a wired laptop) in step (10) (*cf.* Table 5.3). This storage is then to be transported to an air-gapped computer. Options for the transport are USB flash drives, optical media like DVD's, printed paper and scanning (optical character recognition), QR code software, smartcards or contactless via RFID. Depending on the choice of transport, the system would either be more secure or more convenient. Since security is considered to be very important, for this scheme, one will choose the safest variant or use RFID since it is used anyway in this use case. With a private key, one could then hash and sign the VSC signature, which is called VVR from then on, and transport it back to another online machine where the VVR is uploaded on IPFS in step eleven (11) (*cf.* Table 5.3). This terminates the process. The publication of all results on the municipal, cantonal and federal level finalizes the tallying phase [56]. Moreover, a detailed augmented audit trail is shown in figure 5.2.

5.2 Architecture

The design proposed in the last section requires an underlying technical architecture. In the following, an architecture for the POC and the Ballot Tracker is proposed.

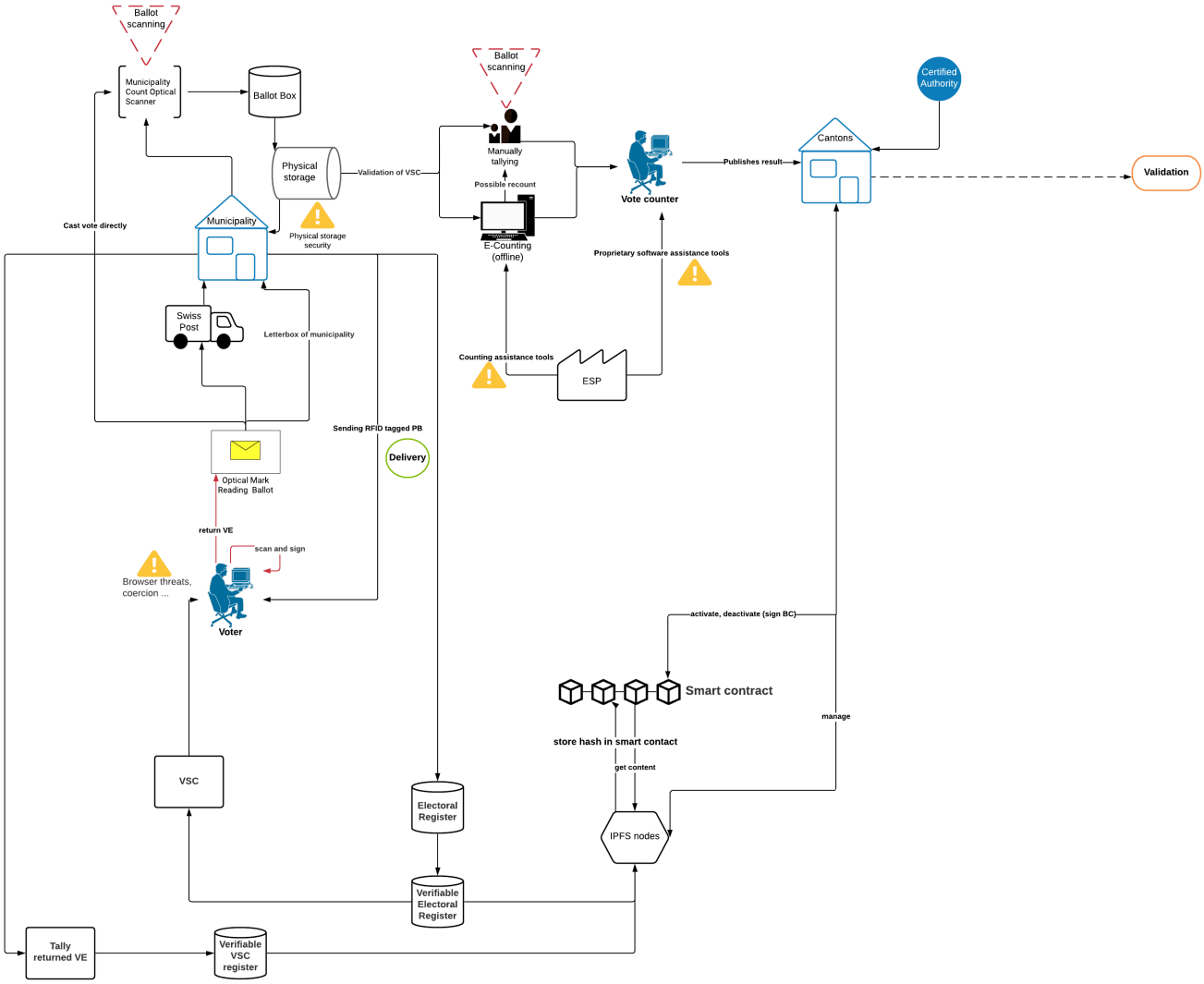

Figure 5.2: Detailed augmented audit trail

5.2.1 POC design

The POC's purpose is to allow the authorities to publish data about the VER and to enable the voters to fetch them. To keep the use case as simple as possible and as exact as necessary, one SPA can combine both of them. Beside that, there are also the login and the identity management side to solve. Thereby, the idea is to cryptographically prove the ownership of an account [64]. This can be done by signing a part of the data with a private key. When the signing of this data generated by the backend is done, the user will be seen as the owner of the public address. According to that, a message-signing-based authentication mechanism can be created by using the user's public address as their identifier [64].

5.2.2 Ballot Tracker design

The ballot tracker design follows the equations mentioned in [70]. The proposed path authentication protocol consists of three phases: a) the system initialization phase, where the required parameters are generated, b) the state update phase, where the tag will be updated and c) the path authentication phase, where the validity verification is realized by the manager [70]. The whole process is mentioned in the steps below (according to [70]):

With the system initialization the system's public parameters are generated for a given security parameter k:

$$parameters = (p, \mathbb{G}, \mathbb{G}_T, g, e, H) \tag{5.1}$$

with p as a large prime number satisfying |p| = k, g as a generator of G of order p, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ representing a bilinear map. Moreover, a cryptographically secure hash function H: $\{0,1\}^* \rightarrow \mathbb{G}$ is chosen by the initializer.

After generating the parameters, the initializer I selects a random element $x_i \in \mathbb{Z}_p$ and computes $Rpk_i = g^{x_i}$ for each reader R_i (with $1 \leq i \leq n$). The public key is Rpk_i and the matching private key is x_i whereas the latter is sent to the readers through a secure channel. The product of all n readers' public keys is then computed by the initializer:

$$PKP = g^{\sum_{i=1}^{n} x_i} = \prod_{i=1}^{n} Rpk_i \tag{5.2}$$

In step 3 two random elements $x_{k_1}, x_{k_2} \in \mathbb{Z}_p$ are chosen by the verification manager M_k ((with $1 \leq k \leq m$)). Additionally, the public-private key pair of M_k is set to $(Mpk_k, Msk_k) = ((Mpk_{k_1} = g^{x_{k_1}}, Mpk_{k_2} = g^{x_{k_2}}), (x_{k_1}, x_{k_2}))$. Finally, the private key Msk_k is sent to M_k through a secure channel.

The initializer chooses a random element $a_{j,0} \in \mathbb{Z}_p$ and computes

$$\begin{aligned} S_{j,1}^{(0)} &= g^{a_{j,0}}, \\ S_{j,2}^{(0)} &= H(ID_j), \\ S_{j,3}^{(0)} &= H(H(ID_j), P_{i_1,i_2,...,i_r;k)}, \\ S_{j,4}^{(0)} &= PKP^{a_{j,0}}, \end{aligned} \tag{5.3}$$

for each tag T_j traveling a valid path (with r general readers $(R_{i_1}, R_{i_2}, ..., R_{i_r})$ and one verification manager M_k.

$$P_{i_1,i_2,...,i_r;k} = \{I, R_{i_1}, R_{i_2}, ..., R_{i_r}, M_k\} \tag{5.4}$$

The ID_j is the identity of the tag T_j. After those computations, the initializer writes the state information $S_{j,t}^{(0)} = 1, 2, 3, 4$, into the memory of T_j. This concludes the initialization phase.

The next phase, the tag state update phase, considers the following simple path as an example:

$$P_{1,2,...,r;k} = I, R_1, R_2, ..., R_r, M_k \tag{5.5}$$

Every tag with identity ID needs to go through this path (for the remaining part of the paper). The equations are according to equation 5.3 and are as follows:

$$\begin{aligned} S_1^{(0)} &= g^{a_0}, \\ S_2^{(0)} &= H(ID), \\ S_3^{(0)} &= H(H(ID), P_{1,2,...,r;k)}, \\ S_4^{(0)} &= PKP^{a_0}, \end{aligned} \tag{5.6}$$

With tag T arriving to the reader R_i, three steps are executed by the latter. First, R_i reads the current state $S_t^{(i-1)}$, $1 \leq t \leq 4$, of T. In the second step $_i$ chooses a random integer $a_{j,0} \in \mathbb{Z}_p$ and computes

$$\begin{aligned} S_1^{(i)} &= (S_1^{(i-1)})^{a_i}, \\ S_2^{(i)} &= (Mpk_{k_1})^{x_i} \times S_2^{(i-1)}, \\ S_3^{(i)} &= (Mpk_{k_2})^{x_i} \times S_3^{(i-1)}, \\ S_4^{(i)} &= \left(\frac{S_4^{(i-1)}}{(S_1^{(i-1)})^{x_i}} \right)^{a_i}, \end{aligned} \tag{5.7}$$

where (g^{x_i}, x_i) is the public-private key pair of R_i and $Mpk_k = (Mpk_{k_1}, Mpk_{k_2}) = (g^{x_{k_1}}, g^{x_{k_2}})$ is the public key of the manager M. In the third step, state R_i updates the state of T with the computed results $S_t^{(i)}$, $1 \leq t \leq 4$.

In the last phase, path autentication is done by the manager M_k at the end of path $P_{1,2,...,r;k}$. To complete the validity verification five steps will be completed. First, M_k reads the actual state $S_t^{(r)}$, $1 \leq t \leq 4$, of T. After that, M_K computes the product of all the readers' public keys in this path

$$PP_{1,2,...,r} = \prod_{j=1}^{r} Rpk_j \tag{5.8}$$

Accordingly,

$$\overline{PP}_{1,2,...,r} = \frac{PKP}{PP_{1,2,...,r}}. \tag{5.9}$$

In the third step M_k checks whether the next equation holds true for:

$$e(S_1^{(r)}, \overline{PP}_{1,2,...,r}) = e(g, S_4^{(r)}) \tag{5.10}$$

If the previous equation holds, tag T has traversed $\mathrm{PP}_{1,2,\ldots,r;k} and M_k$ therefore checks the validity of its path and computes

$$h = \frac{S_2^{(r)}}{(PP_{1,2,\ldots,r})^{x_{k_1}}} \tag{5.11}$$

Should 5.10 not hold, M_k returns $\perp$ and the verification is aborted. In the last step M_k checks again if the equation holds true:

$$\frac{S_3^{(r)}}{(PP_{1,2,\ldots,r})^{x_{k_2}}} = H(h, P_{1,2,\ldots,r;k}) \tag{5.12}$$

If it proves to be true, the path $P_{1,2,\ldots,r;k}$ is considered valid and the process is finalized. The correctness of the last equation is proved by additional equations. This proof can be found in [70].

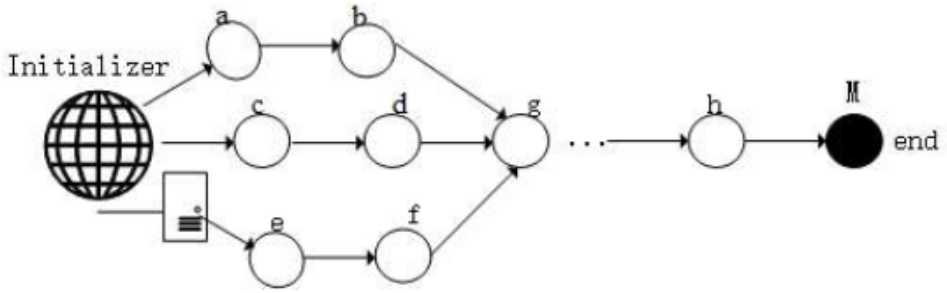

Figure 5.3: Path authentication (Illustration adapted from [70])

The pseudo code for a possible implementation of the Ballot Tracker formulas mentioned above, is presented and described in the Algorithms 1, 2 and 3. It is based on the scheme proposed in [70]. Most path authentication protocols consists of four entities: the tag, the reader(-s), the manager and the initializer of which the latter two have already been described. The tag is attached to an object acting as a transporter to transfer its state among readers. These readers read the stored state in the tag and update them with a new computed state [70]. The paper of [70], as well as the described pseudo code below, are separated in three phases: the system initialization, the tag state update and the path authentication phase.

Algorithm 1: System_initialization

input : m, n, λ, k
output: tag_memory,
(xk1,xk2),($Z_p, x_i, generator1$), $(Mpkk1, Mpkk2), parameters, PKP, valid\_path$

$G \leftarrow$ Bilinear Pairing Group();
$\|p\| \leftarrow$ k;
$Generator1 \leftarrow$ G.generator1();
$Generator2 \leftarrow$ G.generator2();
$G_T \leftarrow$ G.pair(Generator1, Generator2);
$e \leftarrow$ G.pair();
$h \leftarrow$ hash();
$parameters \leftarrow$ (p, G, $G_T, Generator1, Generator2, e, h$)
$PKP \leftarrow 0$;
$R \leftarrow$ array[ ];
$Z_p \leftarrow$ G.order();
for $i = 0$ **to** n **do**
  $R.insert(i)$;
  $x_i \leftarrow Set\ Z_p\ by\ random$;
  $public\_key \leftarrow power(generator1, x_i)$;
  $private\_key \leftarrow x_i$;
  $PKP \leftarrow PKP + public\_key$;
  Send private key to Ri through a secure channel;
end
$M \leftarrow$ array[ ];
$pub\_privkey\_pair \leftarrow$ array[ ];
for $k = 0$ **to** m **do**
  $M.insert(k)$;
  $xk1 \leftarrow Set\ Z_p\ by\ random$;
  $xk2 \leftarrow Set\ Z_p\ by\ random$;
  $Mpkk1 \leftarrow power(generator1, xk1)$;
  $Mpkk2 \leftarrow power(generator1, xk2)$;
  $(Mpkk, Mskk) \leftarrow ((Mpkk1, Mpkk2), (xk1, xk2))$;
  $pub\_privkey\_pair.insert((Mpkk, Mskk))$;
  $PKP \leftarrow PKP + public\_key$;
  Send private key Mskk to the Manager M_k through a secure channel;
end
$tag \leftarrow$ array[ ];
$tag\_memory \leftarrow$ array[ ];
for $j = 0$ **to** λ **do**
  $tag.insert(j)$;
  $valid\_path \leftarrow (Initializer, Readers, Manager)$;
  $ID \leftarrow tag[j]$;
  $aj0 \leftarrow Set\ Z_p\ by\ random$;
  $sj1 \leftarrow power(generator1, aj0)$;
  $sj2 \leftarrow hash(ID)$;
  $sj3 \leftarrow hash(hash(ID, valid\_path))$;
  $sj4 \leftarrow power(PKP, aj0)$;
  $tag\_memory.insert([sj1, sj2, sj3, sj4])$;
end

Algorithm 2: Tag_state_ update

input : System_initialization
output: tag_memory
$pub\_privkey\_pair\_Reader \leftarrow array[]$;
$private\_key\_manager \leftarrow System\_initialization[1]$;
$Z_p \leftarrow System\_initialization[2][0]$;
$x_i \leftarrow System\_initialization[2][1]$;
$generator1 \leftarrow System\_initialization[2][2]$;
$Mpkk1, Mpkk2 \leftarrow System\_initialization[3]$;
$tag\_memory \leftarrow System\_initialization[0]$;
$a_i \leftarrow Set\ Z_p\ by\ random$;
for $i = 0$ **to** tag_memory **do**
  $s1 \leftarrow power(i[0], a_i)$;
  $s2 \leftarrow power(Mpkk1, x_i) * i[1]$;
  $s3 \leftarrow power(Mpkk2, x_i) * i[2]$;
  $s4 \leftarrow power(i[3]/(power(i[0], xi)), a_i)$;
  $pub\_privkey\_pair\_Reader.insert(power(generator1, x_i), xi)$;
end
$Mpkk\_public\_Manager \leftarrow (Mpkk1, Mpkk2)$;
$tag\_memory \leftarrow (s1, s2, s3, s4)$;

Algorithm 3: Path_authentication

input : System_initialization, Tag_state_update
output: Valid path, Invalid path
$tag\_memory \leftarrow Tag\_state\_update$;
$generator1 \leftarrow System\_initialization[2][2]$;
$PKP \leftarrow System\_initialization[5]$;
$x_j \leftarrow System\_initialization[2][0]$;
$path \leftarrow array[]$;
$r \leftarrow n$;
for $j = 0$ **to** r **do**
  $path.insert(power(generator1, x_j))$;
end
$closed\_path \leftarrow PKP/path$;
$e \leftarrow System\_initialization[4][6]$;
if $e(tag\_memory[0], closed\_path) \neq e(generator1, tag\_memory[3])$ **then**
  return false;
end
$h \leftarrow tag\_memory[1]/power(path, initialize[1][0])$;
if $(tag\_memory[2]/(power(path, initialize[1][1])) =$
 $System\_initialization[4][5](h, System\_initialization[6])$ **then**
  return true;
end

The suggested pseudo code could serve as a guidance for an implementation of [70]. An

obvious solution could be the implementation with Python[1], which offers modules for bilinear pairing and has a lot of libraries in general, *e.g.,* for calculation.

5.3 Identity Management (IdM)

Creating IdM systems is a demanding business and has been intensely studied over the last 20 years, without finding a widely accepted solution [88]. For example, Estonia, a pioneer of e-voting, has developed three different solutions for an electronic ID: an ID-card, a mobile-ID and a smart-ID. Currently, they are only using the former two. The third one uses a specific cryptographic scheme where the signature key is split between the mobile device and the server [45]. Concerning Switzerland, a federal subsidized solution named SuisseID has been published in 2010. After the lacking success of the SuisseID the SwissSign Group introduced the SwissID as its replacement in 2017. It has the advantage of being freely available and more easily manageable. Other blockchain-based attempts for a decentralized IdM have been introduced recently. They are, however, directed at a private environment in a Swiss use case [88]. Since there are no possibilities to connect to the SwissID's datapool as a developer, this prototype uses Metamask and its PKI as the IdM.

5.4 Ballot design

The VSC and the printable PB look the same as in figure 5.4. The VSC contains an additional QR code, while the PB contains an RFID tag. The ballots are simple, and a voter can either vote yes or no. For list voting, it can be assumed that the RFID tag is not included on the PB but can be attached to the chosen PB by the voter.

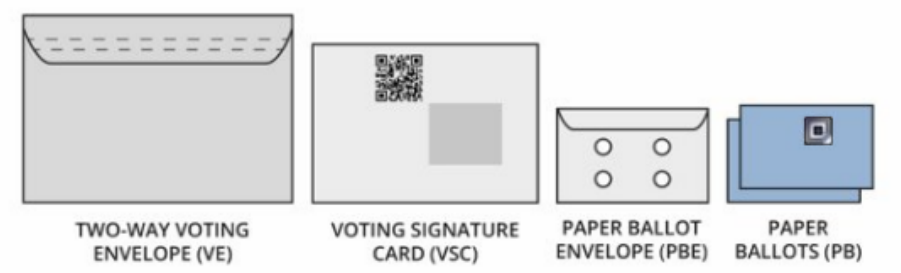

Figure 5.4: Augmented VSC and PB design (illustration adapted and changed from [56])

[1]https://www.python.org/

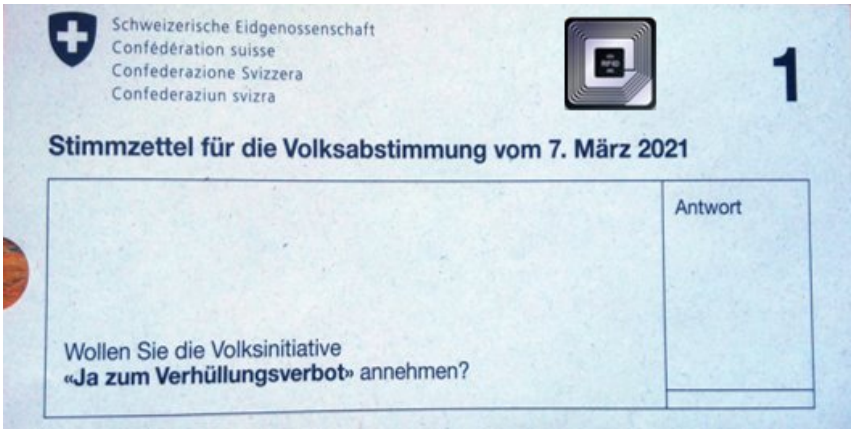

Figure 5.5: New PB design

Chapter 6

Implementation

This section introduces the POC built on the proposed design from the last chapter. The POC can be installed, runs locally, and is divided into a frontend, a backend and a VER part. The POC covers the suggested scheme and the upload from the VER to IPFS as well as the fetching of the VER from the client side. Its implementation is explained by a setup, a components and an application logic section.

6.1 Setup

The implementation of the POC was performed in TypeScript, Javascript and Solidity, and is available on [55]. The POC is divided into two parts. First, a login part for the authority and for the voter, based on the authentication-mechanism from the repository provided in [63]. Secondly, a part to upload files to IPFS from the authority's side. Furthermore, in the Generate ER folder the QR_script.sh can be executed to generate fake citizen data and to create a QR code containing an encrypted VER entry.

6.2 Components

Blockchain: Since this use case focuses on the public part of the Swiss Voting system, only public blockchain platforms have been taken into consideration. From the most popular ones, which are Bitcoin and Ethereum, Ethereum has been used due to its open-source character and developer friendly environment.

Framework: There have been previous projects using an Ethereum development framework as well as efforts to create decentralized applications (Dapp) without using a framework. The used Truffle framework builds a wrapper around the smart contracts enabling their methods to be available as javascript code. Since react is used as the frontend framework, Truffle serves the ABI file to be compatible with typescript. Truffle compiles and,

manages contract deployments and runs automated tests. A part of the Truffle suite is Ganache[1], which enables the execution of local blockchains.

Frontend: The frontend is a single page application (SPA) and created with ReactJS[2]. Web3.js is the used as the interface to the Ethereum blockchain.

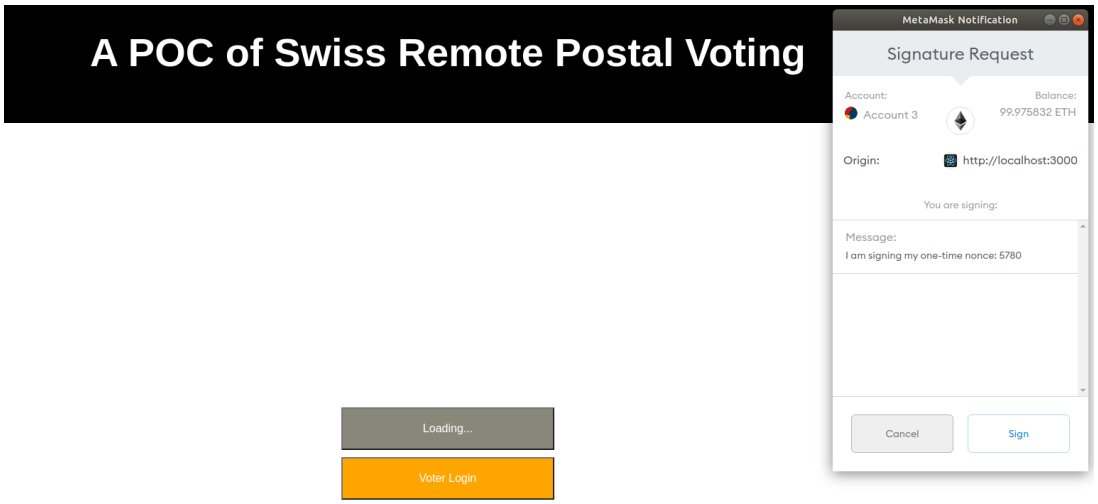

Figure 6.1: Frontend: Login and Signature Request

The frontend is included in the frontend folder of the project. The different subpages of the React SPA are comprised in the src folder together with the abis from the smart contracts, which are also hold in the src folder. The frontend is split into an Administration Frontend and Voter Frontend, where the authorities can upload the file from their machine to IPFS, and the voter can observe the uploaded file.

Backend: The backend is represented by the smart contracts, Node.js[3], expressjs[4] and SQLite[5] to implement the RESTful API. For writing smart contracts, Solidity[6] is applied, as it currently is the most popular smart contract language.

Storage: IPFS is used as the decentralized storage and for file referencing. It stores files and returns the hash to the blockchain which allows them the access the file whenever needed.

Wallet and services: Metamask is a wallet for the browser and enables users to save their Ethereum accounts and their private keys inside the browser. When the frontend needs an interaction with a smart contract, a request can be sent to MetaMask to sign the transaction. MetaMask will forward the request for the user confirmation, and afterwards, the transaction is broadcasted using Infura [62]. Infura is a service and remote node provider which connects the Dapp with Ethereum. Metamask automatically uses Infura

[1]https://www.trufflesuite.com/ganache
[2]https://github.com/facebook/react
[3]https://nodejs.org/en/
[4]https://expressjs.com/
[5]https://www.sqlite.org/index.html
[6]https://docs.soliditylang.org/en/v0.8.3/

in the frontend. The generated API keys are created to connect Web3 and Truffle on their dashboard [62].

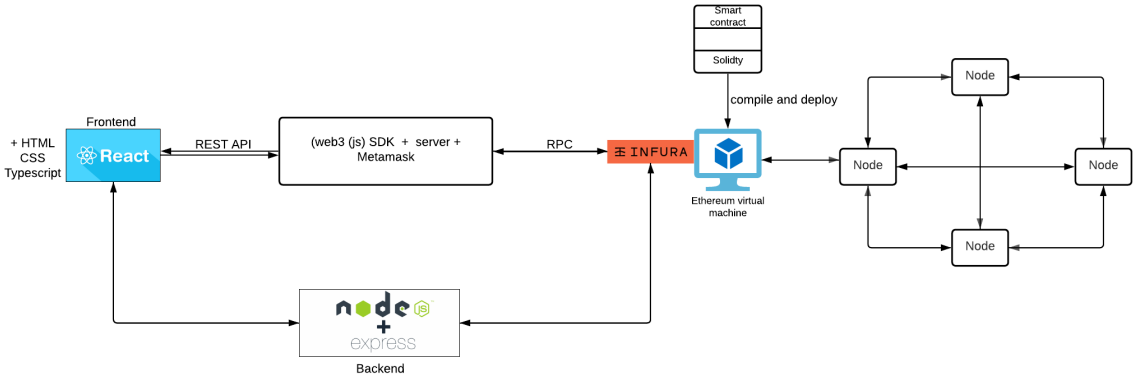

Figure 6.2: Overview application environment

6.3 Application logic

The application logic is splitted into two parts: the MetaMask login and the IFPS upload. The login logic is depicted in figure 6.3 and has been adapted from [64]. In step one (1) (*cf.* Figure 6.3), the user model is modified. A publicAddress and a nonce field are needed, with the latter being set to a random big number. The nonce is changed after each successful login. In step two (2) (*cf.* Figure 6.3), the nonce is generated with the defaultValue function contained in the nonce of the User model, as reflected in figure 6.4. In the next step (3) (*cf.* Figure 6.3), the MetaMask active account is retrieved with web3.eth.coinbase in the handleClick handler function. It checks whether the publicAdress already exists in the backend or not. If it does not exist, a new account in the handleSignup method is created. In step four (4) (*cf.* Figure 6.3), the nonce is signed with the private key associated with the publicAddress through web.personal.sign, which is done in the handlesignMessage function [64]. After the message has been signed successfully by the user, the signature and the publicAddress are sent to the backend through the handleAuthenticate method. With step five (5) (*cf.* Figure 6.3), there is a change back to the backend. The backend receives the request on the /auth route which contains a publicAddress and a signature and has to verify whether this publicAddress has signed the correct nonce [64]. The signature verification happens by proving the ownership of their publicAddress. This is done by using a message msg, which contains the nonce, and the signature. The ecrecover function will then output the public address used to sign the msg with the help of ECC [64]. After a successful authentication, the backend creates a json web token and sends the token back to the client [64]. In the last step (6) (*cf.* Figure 6.3), the nonce is changed due to security reasons.

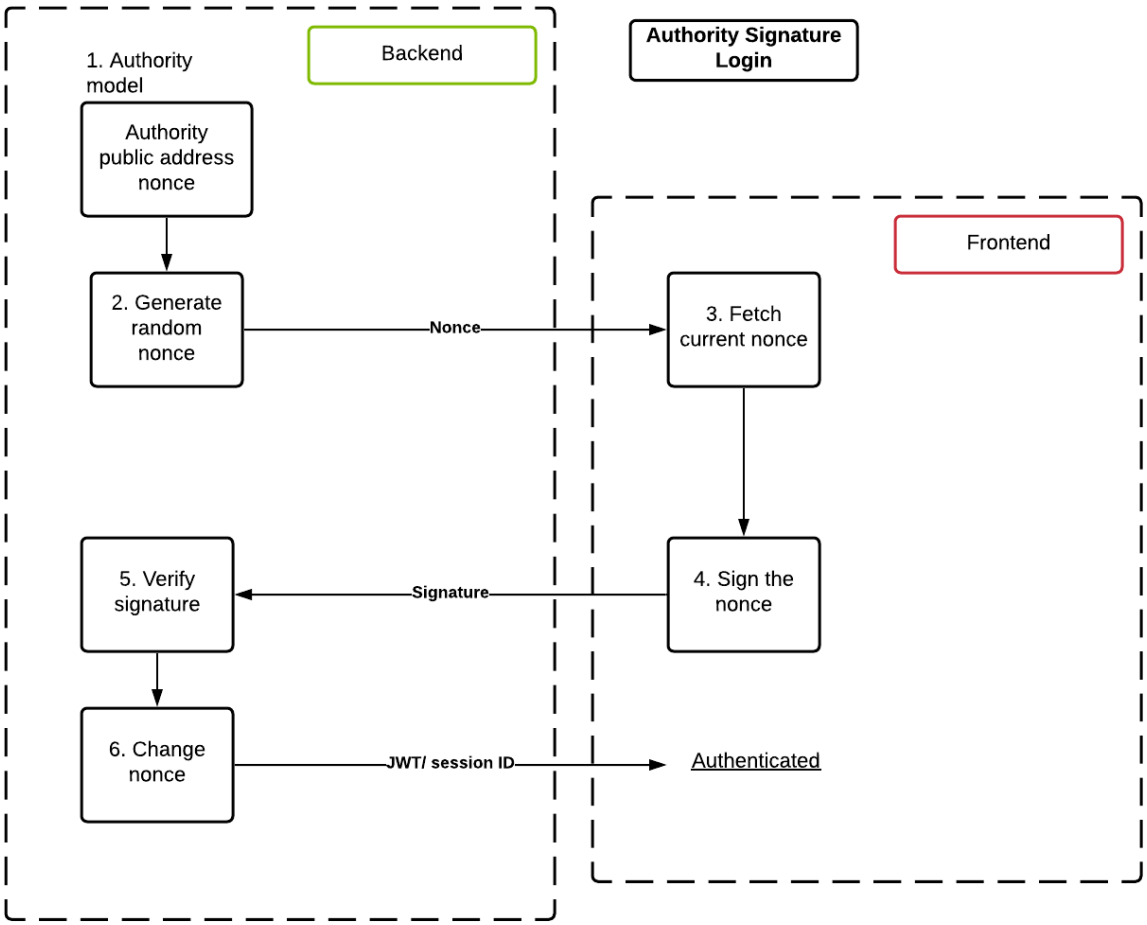

Figure 6.3: Metamask Login Process (Illustration adapted and changed from [64])

```
User.init(
  {
    nonce: {
      allowNull: false,
      type: INTEGER.UNSIGNED,
      defaultValue: (): number => Math.floor(Math.random() * 10000),
    },
    publicAddress: {
      allowNull: false,
      type: STRING,
      unique: true,
      validate: { isLowercase: true },
    },
    username: {
      type: STRING,
      unique: true,
    },
  },
  {
    modelName: 'user',
    sequelize,
    timestamps: false,
  }
);
```

Figure 6.4: User Model (Illustration adapted from [63])

The second logical part is the IPFS upload, which is managed in the Profile folder and Profile.tsx file. Through the frontend the authority is able to chose a file and upload it with IPFS. The IPFS storage returns the hash value for all the saved files on the ethereum blockchain and is managed by the voting authority.

Chapter 7

Evaluation

The implemented POC and suggested RFID-based Ballot Tracker demonstrates the extension of the current paper-based audit trail with added components on its VSC's and PB's, and it also makes use of smart contracts and IPFS. The central aspects of this work, namely the properties from a dedicated privacy, verifiability, usability and scalability perspective, are evaluated and discussed in the following chapter, with the practical deployability in mind, discussing on the implemented PoC. Moreover, the current Swiss RPV scheme is compared to our new scheme.

7.1 Administrative Verifiability

As mentioned before, through AV, election officials are protected from making errors and committing fraud. The proposed RFID-based Ballot Tracking tries to improve and augment the current audit trail with an electronical and a physical component. The extended audit trail tries to answer the question, by adding additional and verifiable signatures in form of a QR code on the VSC, whereas RFID tags on the PB's add more verifiability to the current audit trail without changing it. The use of RFID tags enables the election manager, in this use case the election official with the highest degree of responsibility/duties, to check whether a PB has passed the right path before and after the PB is being counted. As defined in the use case analysis, only time stamps are created, and the counting accuracy is compared in the end. By verifying the taken path, fraud can be reduced to a lower level, while no relation to a certain voter is given. Additionally, errors and fraud are mitigated by letting the voter verify if the VSC is officially coming from the election office. While individual verifiability could not be achieved, it is possible to reach universal verifiability, since the Ballot Tracker sums up all the valid paths. End-to-end verifiability is achieved partially as the voters can verify whether every collected vote is correctly included in the tally by the scanning of the RFID gates [59]. Moreover, receipt-freeness is achieved by not creating a receipt through the whole process. Coercion-resistance in contrast is not possible in this use case. Overall, and in comparison to the current Swiss voting scheme, as can be seen in figure 7.1, the proposed scheme fulfills the same properties as the later one and augments them partly by the universal verifiability

as well as the tallied as collected component from end-to-end verifiability. In the case of universal verifiability, 'partly' means that the voter still has to trust the authority or, more accurately, the election office manager.

	Cryptographic properties						
Voting Schemes	Eligibility	Privacy	Individual Verifiability	Universal Verifiability	End-to-End Verifiability	Receipt Freeness	Fairness
Current Swiss RPV Scheme	x	x				x	
RFID-based Ballot Tracking	x	x		(x)	(x)	x	

Table 7.1: Cryptographic properties of the current and the proposed Swiss RPV Scheme

7.2 Security

The inherent security properties have been defined earlier. The evaluation of the performance of the authentication protocol can be found in [70]. Moreover, several trust assumptions have been made and new threats have been found, as displayed in table 7.2.

7.2.1 Trust model

The used trust model for this use case is based on the following statements:

- The election office, including the election office manager, is trusted.
- ESP's are trusted.
- Metamask is installed on the computer of the authority and the voter side.
- The voting platforms server is trusted [81].
- A fraction of the voters might not be trustworthy [81].
- The client side and the communication channel between the server side and the client side is not trusted [81].
- Given proofs generated by the system, which will be verified by auditors, at least one of the auditors and her technical aids, *i.e.,* software or hardware tools are trusted to function properly [81].
- The camera/scanner is secure which reads the QR code.

7.2.2 Threat events (TE)

The threats model is grouped according to the already known Threat Events (TE) of the Swiss PVPF mentioned in [56]. There are also several additional threats due to our proposed voting scheme, as shown in table 7.2.

TE	Definition
TE1	Third party gateway vulnerabilities: Infura opens the door to third party vulnerabilities and man-in-the-middle attacks
TE2	Browser vulnerabilities
TE3	RFID security: Data tampering, cloning of original tags, privacy violation, eavesdropping, spoofing attack, DoS attack, replay attack, relay attack, untraceability, physical attack (EMP, antenna/complete destruction)
TE4	RFID bandwidth consumption
TE5	RFID monitoring

Table 7.2: Threat events

7.3 Use case

As already mentioned, although the simplest option might be to add and record information directly on a ballot, it is highly important that there is no correlation towards an identifiable voter, and privacy must be guaranteed. Moreover, it must also be obvious to the voter that the additional information cannot be correlated. Thus, the separate authentication of the VCS and the PB containing the QR code and the RFID tag is also still physically separated.

The publication of verifiable information (proving the voters' eligibility through hashing of electoral registers, for example) is important in order to build trust in any digital process [57]. The used Ethereum smart contracts support a broad range of functionality, but also offer vulnerabilities. Those vulnerabilities will not be evaluated here since the smart contract is only a small part of the project. Besides that, it can be seen that the Ballot Tracker guarantees that the PB's are not spoiled by chance, in contrast to the current voting scheme.

7.4 Realworld feasibility

As declared in the beginning, the overarching goal of this thesis is to offer a more verifiable and trackable audit trail for the current Swiss RPV. Therefore, this use case assumes that

the current voting area won't change to an e-voting system soon, and that this use case won't change the whole voting landscape but will add some small electronical and physical features. Especially with regard to the identity management, the addition of one feature to an already working system can lead to several other (hidden) consequences.

Adding the suggested RFID components, the whole measurement equipment, including RFID tagged ballots and RFID gates as well as using IPFS and smart contracts deployments, adds additional costs. The following cost analysis has been made for the city of Zurich, which has 221'000 eligible voters (excluding citizens abroad) [99]. As in previous elections, around 2'500 vote counters are active [98]. The current costs of one election (Sun-)day is assumed to be 600'000 Swiss francs (CHF) according to [86]. 300'000 CHF are due to postal charges, another 200'000 CHF are used as a compensation for the vote counters, and 100'000 CHF are used for the printing of the voting material. Calculating with 0.3 CHF[1] per RFID tagged ballot, one could expect a raise in cost of 221'000 x 0.3 CHF = 66'300 CHF per initiative, referendum or election. Depending on the amount of them, it gets higher in expenses very fast. Additionally, the 12 x 2 = 24 gates have to be purchased once and cost between 2'000 and 15'000 CHF[2], depending on the complexity of the system. Moreover, there is the upload to IFPS. Assuming and as calculated in this use case, the created QR code per voter needs around 30 kB of storage and therefore, the whole storage of generated QR codes would need 66.3 gigabytes of storage. The price for this amount of IPFS storage would therefore be negligible.

Entity	**Amount**	**Costs**
IPFS deployment	221'000 transactions	Price negligible
RFID tagged ballots	221'000 people	66'300 CHF per election
RFID gates	24	48'000 - 360'000 CHF

Table 7.3: Additional election costs for the Canton of Zürich

Besides the economical and technological aspects, the ecological view has to be taken into account as well. At least 50 percent of the current paper used for the PB's is recyclable [98]. When counting the RFID tagged PBs, they have to be removed manually to be separated for recycling. Accordingly, the process becomes more time-consuming. Overall, the costs are raised by 10 to 30 percent per election plus the fix costs of the RFID gate readers in this use case. Assuming that the current voting system is still low in costs being under 3 CHF/voter per election (Sun-)day, the suggested additional equipment would lead to an increase in costs but is still at a low price level. Therefore, and also considering the increase in counting time which could lead to an increase in the compensation for the vote counters, scalability should be arguable.

7.5 Discussion

As we discussed in previous chapters, there is a high amount of different voting schemes. A lot of work has been invested in the past years to improve voting schemes by developing

[1]Price taken from www.digikey.ch
[2]Prices taken from www.identech.ch

new e-voting systems, blockchain-based variants of them or to extend or change the current paper audit trails. Moreover, frameworks have been proposed to compare the different voting schemes regarding their strengths and weaknesses. The difficulties in achieving more verifiability for the voter side arise mainly from the trade-off between verifiability and privacy where the latter is heavily weighted in the Swiss Use Case. Additionally, most of the proposed schemes are working with receipts and are therefore not coercion resistant.

The suggested scheme and prototype still achieves the same properties of the current Swiss RPV while offering small additional features towards more verifiability. Instead of using RFID technology, one could argue to use an electronic delivery of the PB's to eliminate the already mentioned properties and risks of the snail mail delivery. However, this would increase the complexity affecting the delivery and the contents presentation. Also, taking RFID readers and the ongoing discussion about E-Identity into consideration, one could extend the scanning of RFID material in several steps. One such possibility would be to have the RFID tags scanned by a postal worker in the moment of the postal delivery. Moreover, concerning the used RFID technology, another solution and alternative would be to add a separate RFID tagged artifact going through the same stages as the paper ballot. By doing that, only one RFID tag would be used and costs could be kept at a lower level.

Chapter 8

Summary and Conclusions

This work suggested a possible augmentation of the current Swiss RPV, while avoiding a shift towards an e-voting system. It began with an introduction to blockchain (-related topics), showed the trade-off between privacy and verifiability and gave an insight into the current voting landscape of Switzerland. Furthermore, it compared several voting schemes on their different properties and introduced related technologies and theoretical concepts behind this thesis's topic. The analysis of current voting schemes and possible improvements of them have demonstrated that many of these schemes are receipt-based or/and use cryptographic tools. On the contrary, the RFID-based Ballot Tracking keeps the current voting Swiss RPV receipt-free, while making use of Internet of Things (IoT) devices, applying a supply chain approach and blockchain technologies for storage purposes. In its essence, this use case tries to improve the current voting scheme in a rather evolutionary, as opposed to a revolutionary way, by adding small features and hereby avoids fancy cryptographic instruments. Hence, also the shift towards privacy has not been altered and the focus has been laid on changing the verifiability component.

The evaluation has shown that a possible implementation of the Ballot Tracker underlies certain security issues and threats. In addition, the augmented VSC allows the voter the verification of the very. Moreover, the proposed RFID Ballot Tracker functions as an insurance for the voter, as it proves that PB's have taken the very official path step. Besides these facts, the introduction of the RFID technology means an increase in costs, which are assumed to be arguable. Compared to those costs, the created POC, which allows the upload of files to IPFS, serves as a cheap storage for the authorities. While in this use case the sign in for the app has simply been solved with MetaMask using Infura, in practice, IdM is still an ongoing question for which, the Proverum prototype is offering a possible solution.

Overall, one can argue that traditional paper voting in the Swiss RPV is still beneficial although a lot of trust in third parties is required. Most e-voting solutions *e.g.,* from Swiss Post, are still erroneous or/and can not fulfill all the required verifiability properties yet. Furthermore, the proposed scheme contributes to the security of the Swiss RPV by adding a QR code on the VSC's and by appending an RFID tag on the PB's. It however remains arguable how well the proposed scheme can be applied in practice, as it implies additional costs.

8.1 Future work

As shown in the evaluation section, the costs per election are raised around 10 to 30 percent compared to the current scheme. To verify that the proposed scheme and its costs are justifiable, the scheme, however, needs to be applied practically first, to make unintended side-effects evident and to check whether the system is working properly or not.

Combining the Ballot Tracker with the E-Counting tools used in some cantons, the poll workers could scan the VSCs including the QR code and its signature offline and store it together into an offline storage. Besides storing the VVR on IPFS, another option would be to broadcast the signature to a public bulletin board and let everybody verify the counting of all votes to achieve universal verifiability. By doing so, everyone could see the valid pathmark, which means it would then no longer be exclusively available to the election office manager (which otherwise still has to be blindly trusted). Moreover, this use case's POC could be combined with the Proverum prototype. In this prototype, the publishing of the VVR to IPFS could be integrated as well. With the combination of both prototypes, the IdM problem would be solved and the Metamask login could be replaced. At the same time, the user interface could be improved. Detached from this considerations, the current yes/no question scheme can be augmented by adapting the scheme to the election level.

Bibliography

[1] Hyperledger 2019. Identity. https://hyperledger-fabric.readthedocs.io/en/release-1.4/identity/identity.html. Accessed: 10.01.2021.

[2] Claudia Z Acemyan, Philip Kortum, Michael D Byrne, and Dan S Wallach. Usability of voter verifiable, end-to-end voting systems: Baseline data for helios, prêt à voter, and scantegrity {II}. In *2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 14)*, 2014.

[3] Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[4] Syed Taha Ali and Judy Murray. An overview of end-to-end verifiable voting systems. *Real-world electronic voting: Design, analysis and deployment*, 173, 2016.

[5] Seres István András. A blind-signature-based e-voting platform on ethereum. https://github.com/seresistvanandras/evoting. Accessed: 05.01.2021.

[6] Seres István András. Implementing an e-voting protocol with blind signatures on ethereum. https://medium.com/coinmonks/implementing-an-e-voting-protocol-with-blind-signatures-on-ethereum-411e88af044 Accessed: 05.01.2021.

[7] Seres István András. Remote electronic voting: a recap from a blockchain perspective. https://medium.com/@Istvan_A_Seres/remote-electronic-voting-a-recap-from-a-blockchain-perspective-2d726d4fe23b. Accessed: 05.01.2021.

[8] Anastasios Arampatzis. Homomorphic encryption: What is it and how is it used. https://www.venafi.com/blog/homomorphic-encryption-what-it-and-how-it-used. Accessed: 19.11.2020.

[9] Josh Benaloh. Administrative and public verifiability: can we have both? *EVT*, 8:1–10, 2008.

[10] Josh Benaloh, Peter YA Ryan, and Vanessa Teague. Verifiable postal voting. In *Cambridge International Workshop on Security Protocols*, pages 54–65. Springer, 2013.

[11] David Bernhard and Bogdan Warinschi. Cryptographic voting - A Gentle Introduction. In *Foundations of Security Analysis and Design VII*, pages 167–211. Springer, 2013.

[12] Matthew Bernhard, Josh Benaloh, J Alex Halderman, Ronald L Rivest, Peter YA Ryan, Philip B Stark, Vanessa Teague, Poorvi L Vora, and Dan S Wallach. Public evidence from secret ballots. In *International Joint Conference on Electronic Voting*, pages 84–109. Springer, 2017.

[13] Erik-Oliver Blass, Kaoutar Elkhiyaoui, Refik Molva, and Eurecom Sophia Antipolis. Tracker: Security and privacy for rfid-based supply chains. In *In NDSSâ11, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011*. Citeseer, 2011.

[14] Joseph Bonneau, E Felten, A Miller, and S Goldfeder. Bitcoin and cryptocurrency technologies arvind narayanan. *Network Security*, 2016(8):4, 2016.

[15] brother. Tracking in der Logistik: Die Vorteile von RFID, QR- und Barcode. `https://www.brother.de/blog/branchentrends/2020/tracking-in-der-logistik`. Accessed: 2.02.2021.

[16] Tom Burt. Electionguard available today to enable secure, verifiable voting. `https://blogs.microsoft.com/on-the-issues/2019/09/24/electionguard-available-today-to-enable-secure-verifiable-voting/`. Accessed: 5.11.2020.

[17] Tom Burt. Protecting democratic elections through secure, verifiable voting. `https://blogs.microsoft.com/on-the-issues/2019/05/06/protecting-democratic-elections-through-secure-verifiable-voting/`. Accessed: 15.10.2020.

[18] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[19] David Chaum. Blind signature system. In *Advances in cryptology*, pages 153–153. Springer, 1984.

[20] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy*, 2(1):38–47, 2004.

[21] David Chaum, Richard T Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen, Alan T Sherman, and Poorvi L Vora. Scantegrity ii: End-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE transactions on information forensics and security*, 4(4):611–627, 2009.

[22] David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan Sherman, and Poorvi Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy*, 6(3):40–46, 2008.

[23] David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, and Poorvi Vora. Scantegrity: End-to-end voter-verifiable optical- scan voting. *Security Privacy, IEEE*, 6:40 – 46, 06 2008.

[24] David Chaum, Peter YA Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In *European Symposium on Research in Computer Security*, pages 118–139. Springer, 2005.

[25] David Chaum, Peter YA Ryan, and Steve Schneider. A practical voter-verifiable election scheme. In *European Symposium on Research in Computer Security*, pages 118–139. Springer, 2005.

[26] Véronique Cortier, Pierrick Gaudry, and Quentin Yang. How to fake zero-knowledge proofs, again. In *E-Vote-Id 2020-The International Conference for Electronic Voting*, 2020.

[27] California County of Alameda. Alameda County Develops a Radio Frequency Identification (RFID) System to Improve "Chain of Custody" for Voting Equipment during an Election. `https://www.acgov.org/rfid/`. Accessed: 13.1.2021.

[28] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. *European transactions on Telecommunications*, 8(5):481–490, 1997.

[29] Edouard Cuvelier, Olivier Pereira, and Thomas Peters. Election verifiability or ballot privacy: Do we need to choose? In *European Symposium on Research in Computer Security*, pages 481–498. Springer, 2013.

[30] Ashraf Darwish and Maged M El-Gendy. A new cryptographic voting verifiable scheme for e-voting system based on bit commitment and blind signature. *Int J Swarm Intel Evol Comput*, 6(158):2, 2017.

[31] Ashraf Darwish and Maged Gendy. A new cryptographic voting verifiable scheme for e-voting system based on bit commitment and blind signature. *International Journal of Swarm Intelligence and Evolutionary Computation*, 06, 01 2017.

[32] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. *Paul Syverson*, 13, 06 2004.

[33] Verein eCH. eCH E-Government Standards. `https://www.ech.ch/del`. Accessed: 20.01.2021.

[34] Verein eCH. eCH-0220 Bewahrung der Gültigkeit elektronischer Signaturen auf Dokumenten, June 2018.

[35] Verein eCH. eCH-0110 Schnittstellenstandard Abstimmungs- und Wahlresultate Standard, November 2020.

[36] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *International Workshop on the Theory and Application of Cryptographic Techniques*, pages 244–251. Springer, 1992.

[37] g2.com. Best multi-factor authentication (mfa) software.

[38] Craig Gentry. Fully homomorphic encryption using ideal lattices. volume 9, pages 169–178, 01 2009.

[39] Kristian Gjøsteen. Analysis of an internet voting protocol. *IACR Cryptol. ePrint Arch.*, 2010:380, 2010.

[40] Kristian Gjøsteen, Clémentine Gritti, and Kelsey N Moran. Ballot logistics: Tracking paper-based ballots using cryptography. *E-Vote-ID 2020*, page 259, 2020.

[41] Stephen Gossett. How homomorphic encryption could bolser confidence in elections. `https://builtin.com/cybersecurity/electionguard-homomorphic-encryption/`. Accessed: 1.11.2020.

[42] Adrian Gottwald. Wahlsystem der Schweiz. `https://www.vimentis.ch/d/publikation/435/Wahlsystem+der+Schweiz.html`. Accessed: 15.01.2021.

[43] Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

[44] Sue Halpern. Can our ballots be both secret and secure?

[45] Sven Heiberg, Kristjan Krips, and Jan Willemson. Planning the next steps for estonian internet voting. *E-Vote-ID 2020*, page 82, 2020.

[46] IBM. fhe-toolkit-linux, 2020.

[47] IPFS. What is ipfs? `https://docs.ipfs.io/concepts/what-is-ipfs/#decentralization`. Accessed: 15.3.2021.

[48] Natsuki Ishida, Shinâichiro Matsuo, and Wakaha Ogata. Divisible voting scheme. In *International Conference on Information Security*, pages 137–150. Springer, 2003.

[49] MSRC Jarek Stanley. Introducing the electionguard bounty program. `https://msrc-blog.microsoft.com/2019/10/18/introducing-the-electionguard-bounty-program/`. Accessed: 17.10.2020.

[50] Douglas Jones and Barbara Simons. *Broken ballots: Will your vote count?* CSLI Publications Stanford, 2012.

[51] Wen-Shenq Juang and Chin-Laung Lei. A secure and practical electronic voting scheme for real world environments. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 80(1):64–71, 1997.

[52] Atte Juvonen et al. A framework for comparing the security of voting schemes. 2019.

[53] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A systems perspective. In *14th USENIX Security Symposium (USENIX Security 05)*, Baltimore, MD, July 2005. USENIX Association.

[54] Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A systems perspective. *USENIX Security Symposium*, 01 2005.

[55] Christian Killer. provotum/verifiabilitysrpv. `https://github.com/provotum/VerifiabilitySRPV`. Accessed: 10.03.2021.

[56] Christian Killer and Burkhard Stiller. *The Swiss Postal Voting Process and Its System and Security Analysis*, pages 134–149. 09 2019.

[57] Christian Killer, Lucas Thorbecke, Bruno Rodrigues, Eder J. Scheid, Muriel Figueredo Franco, and Burkhard Stiller. Proverum: A hybrid public verifiability and decentralized identity management. *CoRR*, abs/2008.09841, 2020.

[58] Marie-José Kolly. Der Quellcode des E-Voting-Systems ist problematisch, und das hat nicht nur mit Sicherheit zu tun? `https://www.nzz.ch/schweiz/e-voting-der-quellcode-ist-undurchsichtig-sagen-experten-ld.1461406`. Accessed: 15.10.2020.

[59] Robert Krimmer, Melanie Volkamer, Nadja Braun Binder, Norbert Kersting, Olivier Pereira, and Carsten Schürmann. *Electronic Voting: Second International Joint Conference, E-Vote-ID 2017, Bregenz, Austria, October 24-27, 2017, Proceedings*, volume 10615. Springer, 2017.

[60] KU LEUVEN. Co6gc: Introduction to zero-knowledge proofs (part 1). `https://www.esat.kuleuven.be/cosic/blog/co6gc-introduction-to-zero-knowledge-proofs-1/`. Accessed: 12.03.2021.

[61] Yi Liu and Qi Wang. An e-voting protocol based on blockchain. *IACR Cryptol. ePrint Arch.*, 2017:1043, 2017.

[62] soliditydeveloper Markus Waas. The big picture of solidity and blockchain development in 2020. `https://soliditydeveloper.com/solidity-overview-2020`. Accessed: 12.12.2020.

[63] AMAURY MARTINY. login-with-metamask-demo. `https://github.com/amaurym/login-with-metamask-demo`. Accessed: 28.02.2021.

[64] AMAURY MARTINY. One-click Login with Blockchain: A MetaMask Tutorial. `https://www.toptal.com/ethereum/one-click-login-flows-a-metamask-tutorial`. Accessed: 11.03.2021.

[65] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.

[66] René Nyffenegger. Blockchain [bitcoin]. `https://renenyffenegger.ch/notes/development/Crypto-Currencies/Bitcoin/blockchain`. Accessed: 23.11.2020.

[67] Bhumika Patel, Purvi Tandel, and Slesha Sanghvi. *Efficient Ballot Casting in Ranked Based Voting System Using Homomorphic Encryption*, pages 565–576. 07 2019.

[68] Selwyn Piramuthu. Protocols for rfid tag/reader authentication. *Decision Support Systems*, 43(3):897–914, 2007.

[69] Parlamentarische Verwaltungskontrolle (PVK). Elektronische Auszählung von Stimmen (E-Counting) Bericht der PVK zuhanden der Geschäftsprüfungskommission des Nationalrates, February 2017.

[70] Yuyin Qian, Peng Zeng, Zuming Shen, and Kim-Kwang Raymond Choo. A lightweight path authentication protocol for rfid-based supply chains. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1297–1302. IEEE, 2018.

[71] Michael J Radwin and Phil Klein. An untraceable, universally verifiable voting scheme. In *Seminar in Cryptology*, pages 829–834, 1995.

[72] BBC News Reality Check Team. Us election 2020: Do postal ballots lead to voting fraud? `https://www.bbc.com/news/world-us-canada-53353404`. Accessed: 15.10.2020.

[73] Ronald L Rivest. The threeballot voting system. 2006.

[74] Ronald L Rivest and Warren D Smith. Three voting protocols: Threeballot, vav, and twin. *USENIX/ACCURATE Electronic Voting Technology (EVT 2007)*, 2007.

[75] Geir Røsland. Remote electronic voting. *Hovedoppgave, University of Bergen, Norway*, 2004.

[76] P. Y. A. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia. PrÃt Ã voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security*, 4(4):662–673, 2009.

[77] Peter YA Ryan. A variant of the chaum voter-verifiable scheme. In *Proceedings of the 2005 Workshop on Issues in the Theory of Security*, pages 81–88, 2005.

[78] Roy Saltman. *The history and politics of voting technology: In quest of integrity and public confidence*. Springer, 2006.

[79] Michael Schläpfer. *Secure end-to-end communication in remote internet voting*. PhD thesis, ETH Zurich, 2016.

[80] Michael Schmid and Andreas Grünert. Blind signatures and blind signature e-voting protocols. *University of Applied Science Biel*, 2502.

[81] Scytl. Scytl sVote – Complete Verifiability Security Proof Report, 2018.

[82] SSH Communications Security. Pki - public key infrastructure. `http://web.archive.org/web/20210308205012/https://www.ssh.com/pki/`. Accessed: 12.02.2021.

[83] TECHSPOT Shawn Knight. Microsoft's electionguard aims to boost the security and verifiability of elections. `https://www.techspot.com/news/79959-microsoft-electionguard-aims-boost-security-verifiability-elections.html`. Accessed: 17.10.2020.

[84] Rakesh Shrestha and Shiho Kim. *Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities*. 01 2019.

[85] Rodrigo Silva. *Proposal for Application to Vote Verification with Blockchain Technology: The Approach of an End-to-End Verifiability Model to the Estonian Internet Voting System*. PhD thesis, 05 2020.

[86] SRF. Der günstige Preis der direkten Demokratie. `https://www.srf.ch/news/schweiz/der-guenstige-preis-der-direkten-demokratie`. Accessed: 24.02.2021.

[87] Thales. Iso 14443 contactless card standard. `https://www.thalesgroup.com/en/markets/digital-identity-and-security/technology/iso14443`. Accessed: 24.02.2021.

[88] Lucas Thorbecke. Decentralized identity management for swiss federalism. Master's thesis, University of Zurich, Binzmühlestrasse 14, CH-8050 Zürich, Switzerland, 3 2020.

[89] Alex Thornton. What is electionguard? `https://news.microsoft.com/on-the-issues/2020/03/27/what-is-electionguard/`. Accessed: 15.11.2020.

[90] Heather Walker. E-signatures in elections: The key to preventing voter fraud? `https://www.cryptomathic.com/news-events/blog/e-signatures-in-elections-the-key-to-preventing-voter-fraud`. Accessed: 3.11.2020.

[91] Lionel Walter. Umfrage bezüglich elektronische Ergebnisermittlungssysteme, Aug 21, 2019.

[92] Hongbing Wang, Yingjiu Li, Zongyang Zhang, and Yunlei Zhao. Efficient tag path authentication protocol with less tag memory. In *International Conference on Information Security Practice and Experience*, pages 255–270. Springer, 2016.

[93] Shiyuan Wang, Divyakant Agrawal, and Amr Abbadi. Is homomorphic encryption the holy grail for database queries on encrypted data? 05 2012.

[94] Wikipedia. Benaloh cryptosystem. `https://en.wikipedia.org/wiki/Benaloh_cryptosystem`. Accessed: 22.11.2020.

[95] Mark Will. *A guide to homomorphic encryption*, pages 101–127. 12 2015.

[96] Mark A. Will and Ryan K.L. Ko. Chapter 5 - a guide to homomorphic encryption. In Ryan Ko and Kim-Kwang Raymond Choo, editors, *The Cloud Security Ecosystem*, pages 101 – 127. Syngress, Boston, 2015.

[97] wombat VOTING SYSTEM. How to vote. `https://wombat.factcenter.org/how-to-vote`. Accessed: 15.10.2020.

[98] Limmattaler Zeitung. Tausende Wahlhelfer und mehr als 90 Tonnen Papier. `https://www.limmattalerzeitung.ch/limmattal/tausende-wahlhelfer-und-mehr-als-90-tonnen-papier-ld.1391862`. Accessed: 08.03.2021.

[99] Stadt Zürich. Abstimmungen & Wahlen. https://www.stadt-zuerich.ch/prd/de/index/bevoelkerungsamt/umziehenmelden/abstimmungen---wahlen.html. Accessed: 07.03.2021.

Abbreviations

AV	Administrative Verifiability
BC	Blockchain
CHF	Swiss francs
Dapp	Decentralized application
DHM	Diffie-Hellman-Merkle
EC	E-Counting
ECC	Elliptic Curve Cryptographie
EPC	Electronic Product Code
ESP	External Service Providers
ER	Electoral Register
FHE	Fully homomorphic encryption
FR	Functional Requirements
GUI	Graphical user interface
HE	Homomorphic encryption
HMAC	Hashed message authentication code
IdM	Identity Management
IoT	Internet of Things
MFA	Multi-factor authentication
NFR	Non-Functional Requirements
OVN	Open Vote Network
P2P	Peer-to-Peer
PB	Paper Ballot
PBB	Public bulletin boards
PBE	Paper Ballot envelope
PKI	Public key infrastructure
POC	Proof-of-concept
PRF	Pseudo-random functions
PV	Public Verifiability
PVPF	Postal Voting Process Flow
QR	Quick response
RFID	Radio-frequency identification
RPV	Remote Postal Voting
RPC	Remote procedure calls
RSA	Rivest-Shamir-Adleman
SPA	Single page application
TE	Threat Events

TOR	The Onion Router
UUID	Universally unique identifier
VE	Two-way voting envelope
VER	Verifiable electoral register
VSC	Voting Signature Card
ZKP	Zero Knowledge Proofs

List of Figures

List of Tables

Appendix A

Installation Guidelines

To run the POC one has to clone the repository

`https://github.com/provotum/VerifiabilitySRPV.git`

to a chosen location on the computer. The requirements to be able to run the POC are as follows:

- Node.js 12.0.0 or newer
- Node Package Manager (NPM) 6.0.0 or newer
- Yarn 1.22.10 or newer

Moreover, to make the sign in and the IPFS upload work, the MetaMask browser extension has to be configured with Ganache, a local blockchain for testing purposes. To get started with Ganache you can visit `https://www.trufflesuite.com/docs/ganache/quickstart`. To add MetaMask and connect it with Ganache please follow the instructions on `https://www.trufflesuite.com/docs/truffle/getting-started/truffle-with-metamask`.

From the POC folder of the repository the following commands can be executed on the command line to set up the frontend and backend as well as the root dependencies and to finally run the POC:

```
yarn install
cd packages/backend && yarn install
cd ../frontend && yarn install
cd ../..
yarn start
```

Appendix B

Contents of the CD

The attached CD contains the following items:

- Thesis: The Thesis folder contains the compiled latex output, *i.e.,* the PDF version of this thesis.
- Latex-Source: This folder holds the source files of this report (including graphics).
- POC: The POC folder contains the source code of the implemented prototype.