On Remote and Voter-Verifiable Voting

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

Roberto Samarone dos Santos Araújo
aus Fortaleza, Brasilien

Referenten:	Prof. Dr. Johannes Buchmann Prof. Dr. Peter Y. A. Ryan
Tag der Einreichung:	12. August 2008
Tag der mündlichen Prüfung:	25. September 2008

Darmstadt, 2008
Hochschulkennziffer: D 17

To my Parents

Acknowledgements

I would like to especially thank:

Prof. Dr. Johannes Buchmann for accepting me as his student, for his support, and for the guidance of this Thesis.

Prof. Dr. Peter Y. A. Ryan for the time that I was visiting his group and for the everlasting discussions.

Dr. Jacques Traoré, Prof. Dr. Ricardo F. Custódio, and Prof. Dr. Jeroen van de Graaf, my other collaborators, for the valuable discussions.

Prof. Dr. Jintai Ding, Dr. Marc Fischlin, and Prof. Dr. Poorvi Vora for their important suggestions and remarks.

Dr. Evangelos Karatsiolis and Dominique Schröder for reading this work and for providing helpful comments. Michael Schneider for helping me with the preparation of the German abstract.

Maike Ritzenhofen, Christina Lindenberg, and Dr. Katja Schmidt-Samoa for their help in different aspects.

Marita Skrobic for her support and help.

My parents who encouraged me and were patient during my research abroad.

God for giving me these years of research in Germany where I could learn many things in all areas of my life.

Publications

(1) Roberto Araújo and Peter Y. A. Ryan. Improving the Farnel voting scheme. In Robert Krimmer and Rüdiger Grimm, editors, *Electronic Voting*, volume 131 of *LNI*, pages 169–182. GI, 2008.

(2) Roberto Araújo, Sébastien Foulle, and Jacques Traoré. A practical and secure coercion-resistant scheme for remote elections. In David Chaum, Miroslaw Kutylowski, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Frontiers of Electronic Voting*, number 07311 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. To appear on Towards Trustworthy Election Systems book. David Chaum, Ron Rivest, Markus Jakobsson, Berry Schoenmakers, Peter Ryan, Josh Benaloh, and Mirek Kutylowski, editors.

(3) Lucie Langer, Axel Schmidt, and Roberto Araújo. A pervasively verifiable online voting scheme. In Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, and Christian Scheideler, editors, *GI Jahrestagung (1)*, volume 133 of *LNI*, pages 457–462. GI, 2008.

(4) Roberto Araújo, Ricardo Felipe Custódio, and Jeroen van de Graaf. A verifiable voting protocol based on Farnel. IAVoSS Workshop On Trustworthy Elections (WOTE2007), Ottawa, Canada, June 2007. To appear on Towards Trustworthy Election Systems book. David Chaum, Ron Rivest, Markus Jakobsson, Berry Schoenmakers, Peter Ryan, Josh Benaloh, and Mirek Kutylowski, editors.

(5) Stefan G. Weber, Roberto Araújo, and Johannes Buchmann. On coercion-resistant electronic elections with linear work. In *2nd Workshop on Dependability and Security in e-Government (DeSeGov 2007) at 2nd Int. Conference on Availability, Reliability and Security (ARES'07)*, pages 908–916. IEEE Computer Society, 2007.

(6) Roberto Araújo, Augusto Devegili, and Ricardo Custódio. Farnel: Um protocolo criptográfico para votação digital. (in Portuguese). In *Proceedings of II Workshop em Segurança de Sistemas Computacionais*, Búzios, Rio de Janeiro, Brazil, June 2002.

Zusammenfassung

In modernen Gesellschaften sind Wahlen ein wichtiges Utensil. Sie sind ein fundamentaler Bestandteil der Demokratie. Dieser Mechanismus muss immer fair und korrekt ablaufen. Bedrohungen dieses Wahlprozesses könnten diese Anforderungen beeinträchtigen. Im konventionellen Wahlverfahren beispielsweise könnte ein Betrüger die Wahlurne manipulieren. Beim *Distanzwahl* gibt es aufgrund der unkontrollierbaren Umgebung mehr Bedrohungen. In dieser Arbeit schlagen wir sichere Wahlverfahren mit wichtigen Eigenschaften vor.

Die Prüfbarkeit (Verifiability) des Wählers ist eine neuartige Eigenschaft, die von aktuellen Wahlsystemen bereitgestellt wird. Sie ermöglicht es dem Wähler festzustellen, dass seine Stimme fehlerfrei in der Abrechnung berücksichtigt wurde. Diese Systeme erlauben bessere Sicherheitsgarantien als konventionelle Systeme, da Wähler die Möglichkeit haben Verfälschungen und Fehler bei der Auszählung selbst festzustellen. Meist basieren sie auf Verfahren der Kryptographie, um dieses Merkmal bereitzustellen.

Wir stellen ein neues Konzept der Wahlurne vor, das auf dem Farnel Ansatz basiert. Dabei erhalten Wähler beim Wahlvorgang einen oder mehrere zufällig gewählte Belege über ihre Wahl. Dies erlaubt eine andere Auffassung der Wähler-Verifiability, in der der Wähler eine oder mehrere zufällig gewählte Stimmen verifizieren kann. Dieser Ansatz liefert einige interessante Eigenschaften: das Wahlgeheimnis ist von Beginn an gewährleistet und muss nicht durch Mischen bei der Auswertung erzeugt werden. Des weiteren schwächt es Angriffe durch Randomisierung ab, die anderen *voter-verifiable schemes* anhaften.

Aufbauend auf der neuen Wahlurne stellen wir drei neue voter-verifiable, papierbasierte Wahlsysteme vor. Eines davon kommt ohne Verwendung der Kryptographie aus und ist daher leichter verständlich für den durchschnittlichen Wähler. Die anderen beiden benötigen lediglich ein einziges kryptographisches Primitiv und liefern bessere Sicherheitsgarantien. Zusätzlich zu diesen Systemen stellen wir Erweiterungen des Threeballot Protokolls und des Randell-Ryan Schemas vor. Beide sind voter-verifiable und machen keinen Gebrauch von Kryptographie. Allerdings können sie weniger Garantien leisten als kryptographie-basierte Systeme.

Die gezeigten voter-verifiable schemes berücksichtigen eine kontrollierte Umgebung wie etwa die bekannten, konventionellen Wahlen. Distanzwahl, beispielsweise per Internet, hat viele Vorteile gegenüber diesen, und ist darüber hinaus sehr anspruchsvoll. Obwohl die existierende Technologie solche Wahlen ermöglicht könnte gibt es eine Menge verbundener Bedrohungen wie den Zwang bei der

Stimmabgabe oder den Verkauf der Wahlstimme. Daher geben wir auch ein System zur Internet-Wahl an. Unser Vorschlag schwächt Angriffe per Zwang und ist im Vergleich zu bisherigen Systemen effizienter bei Wahlen in großem Rahmen.

Abstract

Voting is an important tool for modern societies. It is fundamental for the democracy. This mechanism must be fair and accurate. However, threats intrinsic in the voting process may compromise these requisites. In conventional voting, dishonest talliers may corrupt the ballot box, for instance. Remote voting has more threats due to the uncontrolled environment. In this work, we propose secure protocols for polling station and remote voting with attractive properties.

Voter-verifiability is a novel security feature provided by recent voting systems. It allows voters to confirm that their votes are accurately counted in the tally. These systems provide better security guarantees than the conventional ones as voters are able to detect corruptions of votes and counting errors. They usually rely on cryptography to implement this feature.

We introduce a novel concept of ballot box based on the Farnel approach. It provides voters, when they cast their votes, with one or more random selected receipts. This allows a different notion of voter-verifiability in which the voter may verify one or more random votes. The idea has a number of attractive features: ballot secrecy is achieved up front and does not have to be provided by anonymising mixes during tabulation. Also, it mitigates randomization attacks that are inherent in some voter-verifiable schemes.

Based on the new ballot box, we introduce three new voter-verifiable paper-based schemes. One of these schemes does not employ cryptography and thus can be more easily understood by the average voter. The others require just a single cryptographic primitive and achieve better security guarantees. In addition to these schemes, we propose improvements to the Threeballot and Randell-Ryan voting protocols. These schemes are voter-verifiable and do not employ cryptography. However, they ensure less guarantees than cryptographic based systems.

The voter-verifiable schemes given consider a controlled environment as conventional voting. Remote Internet voting, however, has many benefits over the conventional ones and is challenging. Although the existing technology render possible such voting, it has a number of associated threats as coercion and vote selling. We also propose a new scheme for remote Internet voting. The proposal mitigates coercive attacks and is more efficient for large scale voting than the previous solutions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Voting systems have been used since the ancient times and are fundamental for the modern democratic societies. Early systems, though, did not provide security guarantees as those existing today. Before printed ballots became a standard, for instance, votes were issued using tickets provided by political parties or even in voice [55] compromising the voter privacy. The necessity of secure systems, however, became evident and motivated the design of new systems as soon as frauds or their risks arose. Frauds were one of the reasons for the adoption of the Australian secret ballot in 19*th* century [55, 46]. More recently, voting machines had to be improved due to their vulnerabilities [62]. Besides security factors, the advance of the technology has also contributed to the adoption of new systems. Computers have replaced paper ballots in some countries as Brazil. Remote voting via Internet has been used as alternative to conventional methods in Estonia [33].

Traditional paper-based voting is simple and effective. Voters choose their candidates in the privacy of a voting booth and cast their votes into a ballot box. At the end of the voting period talliers open the ballot box and count all votes. This voting system, though, requires trust in the tabulation process. Because manipulations cannot be detected, adversaries (e.g. dishonest officials) may alter, replace, or even spoil votes in order to compromise the final results. In addition, this voting is susceptible to other threats such as chain voting [47, 57].

Differently from systems traditionally used, modern voting schemes such as Prêt-à-Voter [83, 22] and PunchScan [77] require less trust in the systems' components or in the officials. These schemes afford high degree of security by providing voters strong evidences that their votes were accurately tabulated. In particular, voters obtain protected receipts corresponding to their votes and use them to verify their votes are included in the final count (voter-verifiability). Although associated to the voters' votes, the receipts do not leak any information about the candidates chosen. This way, even if voters are prepared to cooperate with coercers or vote bribers, they cannot convince them about their votes.

In order to achieve security and to implement voter-verifiability, modern voting schemes rely on cryptography. Such technology makes the security of these systems comparable or even better than traditional paper-based voting. However, whereas cryptography can be understood by experts, these mechanisms are not easily grasped by voters who cannot comprehend systems' details. Making voting systems understandable, at least for the average voters, let voters trust on their security properties and to accept them more easily.

With the goal of making schemes more simple while preserving the voter-verifiability property, Randell-Ryan [78] and Rivest [80] introduced paper-based schemes that do not rely on cryptography. These schemes can be more easily understood by the voters. However, they do not achieve the same levels of assurance as cryptographic based systems. In the scheme proposed in [80], for example, the ballot secrecy is not perfect and it may reveal statistical indications of voting results before the voting ends.

In this thesis we first propose a voter-verifiable scheme that does not employ cryptography. Our proposal does not expose partial voting results as in [80]. In addition, it differs from the previous solutions by employing a new variant of voter-verifiability. That is, a voter instead of receiving a receipt corresponding to her own vote, she obtains one or more random receipts. In order to implement this, we introduce a novel concept of ballot box. Instead of just receiving a ballot as usual, the new ballot box provides the voters, when their cast their votes, with copies of randomly selected, previous cast votes as receipts. This idea has a number of attractive features. First, ballot secrecy is achieved up front and does not have to be provided by anonymizing mixes during tabulation. Moreover, any fears that voters might have that their votes are not truly concealed in their encrypted receipts are mitigated. Lastly, it helps to prevent randomization attacks.

Apart from being simple for the voters, our initial solution does not afford assurances as those present in cryptographic based schemes. Indeed, the design of voting systems without cryptography while achieving high degree of security is hard as former solutions showed. This way, we introduce two novel voting schemes that employ cryptography. In order to keep the schemes simple for the voters, the proposals make minimal use of cryptography; they require just a commitment scheme. One of the cryptographic schemes follows the same model as our initial scheme. Though, it has better security properties. The other one is a simplification of the first cryptographic scheme.

In addition to these new schemes, we explore ways to introduce the new concept of ballot box into the Threeballot and Randell-Ryan schemes in order to improve them. Our solution for the former aims at solving its information leakage problem where indications of the results can be obtained through the receipts. The improvements for the latter scheme increase its security assurances.

The paper-based voter-verifiable voting schemes proposed in this work consider a physical voting environment. That is, a place controlled by voting officials where the voters go to cast their votes. However, the interest in conducting remote voting over the Internet has increased recently. Countries as Estonia [33] and Switzerland [39] have performed such voting. In these voting, instead of being restricted to controlled places, voters are free to cast their votes over the Internet using any connected computer and places of their convenience.

Remote Internet elections have many advantages over elections that require physical environment. As voters have the possibility to vote using any computer connected to the network and may perform this, for example, from the comfort of their residences, it affords more convenience [36]; as consequence, such voting is more attractive for voters and thus may increase voter participation. In addition, voting results are computed faster and these voting are supposed to cost much less than the polling station ones.

Despite its benefits, Internet voting is susceptible to coercion and vote-selling. Because voters vote from uncontrolled places, coercers and vote buyers can easily influence them to vote for their candidates. A coercer could, for example, observe his victims while they are voting to ensure that they are following his instructions. Moreover, coercer and vote-buyers may automatize attacks to reach a large number of voters. As stated by Jefferson et al. [52], "the Internet can facilitate large scale vote buying by allowing vote buyers to automate the process". Although these are realistic threats that would violate the voter's privacy, they are mostly ignored in practice. In the Internet voting conducted in Switzerland [39], for example, these threats are not considered.

Former Internet voting schemes deal with coercion and vote-selling through the idea of preventing voters from making or obtaining any information that documents their votes. This way, since voters cannot produce any evidence about their votes and transfer them over Internet, they cannot be coerced to vote for a certain candidate or sell their votes. Although this idea helps preventing voters from revealing their votes as long as adversaries are not watching the voter while voting, it is not sufficient to defeat other coercive attacks. Recently, Juels, Catalano, and Jakobsson [60] introduced a more complete notion regarding coercive attacks in Internet voting called coercion-resistance. In their notion, Juels et al. consider that voters may obtain information about their votes and also that they may be forced to abstain from voting, to reveal secret information used for voting, or to cast random votes. The coercion-resistance notion is the strongest known property that a voting scheme can fulfill to achieve security in remote Internet voting scenarios.

Juels et al. also presented the first scheme that satisfies the coercion-resistance property. The scheme provides better security assurances than previous proposals, but it has an intrinsic drawback: "the overhead for the tallying authorities is quadratic in number of voters" [60]. The proposal, thus, can only accomplish small scale elections. Smith [91] and Weber et al. [95] presented schemes based on Juels et al.'s ideas with linear overhead, but their schemes are insecure in the sense of coercion-resistance.

With the goal of satisfying the coercion-resistant property while avoiding inefficiency, we also introduce in this thesis a new coercion-resistant scheme for remote Internet voting. Our proposal follows the ideas of Juels et al. However, it has linear work factor and can be used in large scale elections.

Organization

This work is organized as follows. In Chapter 2 we present the cryptographic primitives necessary to accomplish our proposals. Most of these primitives are employed in the new coercion-resistant voting scheme. After that, in Chapter 3, we introduce the novel concept of ballot box. Next, we give the voter-verifiable schemes in Chapter 4. These schemes are paper-based and employ the new ballot box. The new coercion-resistant voting scheme is presented in Chapter 5. Finally, we discuss future work in Chapter 6.

Chapter 2

Cryptographic Primitives

In this chapter we present the cryptographic primitives which are used in the coercion-resistant voting scheme of Chapter 5. Its security relies on computational assumptions which we also recall in this chapter. Some of these primitives are also necessary to the voter-verifiable schemes that we present in Chapter 4.

2.1 Bulletin Boards

Modern voting schemes rely on bulletin boards to publish their public information (e.g. encrypted votes). This mechanism performs as a broadcast channel that can be read by everyone and has memory to store data. However, only authorized parties are able to write on it and all information written cannot be deleted or modified anymore. The board also adds a timestamp to each data it receives. This communication model for voting was first presented by Benaloh et al. [9, 26] and helps achieving verifiability in modern voting schemes.

Bulletin boards are normally controlled by more than one party in a distributed setting so that a certain number of corrupted parties cannot compromise the board. This way, they can be implemented via a Byzantine agreement scheme, such as the proposals of Cachin et al. [16] or Lindell et al. [66].

Voting schemes may use one or more bulletin boards for different purposes. For example, a bulletin board publishes information used in the set up phase of a voting, another stores votes during the voting, and a third publishes information processed in the tally. Also, variants of the original communication model are possible. The scheme introduced by Juels et al. [60], for example, does not require authentication to write information on the board during the voting.

For the cryptographic primitives presented next as well as for the voting protocols introduced in Chapters 4 and 5, we consider that public communications are performed through bulletin boards.

2.2 Computational Assumptions

The voting scheme of Chapter 5 relies on two known and reasonable computational assumptions that we recall below.

The Decision Diffie-Hellman (DDH) Assumption. We give verbatim the definition by Katz-Lindell [61]. See also Dan Boneh [11].

Let $\mathcal{G}$ be a polynomial-time algorithm that, on input 1^n, outputs a cyclic group $\mathbb{G}$, its order q, and a generator $g \in \mathbb{G}$. The DDH problem is hard relative to $\mathcal{G}$ if for all probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function *negl* such that:

$$\left|Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]\right| \leq negl(n)$$

where in each case the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs $(\mathbb{G}, q, g)$, and then random $x, y, z \in \mathbb{Z}_q$ are chosen.

The LRSW Assumption. This assumption was introduced by Lysyanskaya, Rivest, Sahai, and Wolf [67] and holds for generic groups. It is described as:

Let $\mathbb{G}$ be a cyclic group of prime order q where the decision Diffie-Hellman assumption is hard, $g, g^x, g^y \in \mathbb{G}_q$, and O be an oracle. The LRSW assumption is as follows: by querying O with $r \in \mathbb{G}_q$, it returns (a, a^{yr}, a^{x+rxy}) for $a \in_R \mathbb{G}_q$. After a number of t queries $r_1, r_2, \ldots, r_t$, it is hard to generate (a, a^{yz}, a^{x+zxy}) for $z \notin r_1, r_2, \ldots, r_t$ and $a \neq 1$.

2.3 The El Gamal Public Key Cryptosystem

The El Gamal scheme is a probabilistic public-key cryptosystem due to Taher El Gamal [38]. This cryptosystem is secure under the discrete logarithm and Diffie-Hellman problems. Though, it is possible to base its security on the decision Diffie-Hellman assumption in order to achieve a semantically secure version of this cryptosystem.

We present below a variant of the El Gamal algorithm, called M-El Gamal, proposed by Juels, Catalano, and Jakobsson [60] that is secure under DDH assumption. It uses a subgroup of order q of a cyclic group $\mathbb{G}$.

Parameters and Key Generation

In order to define the group $\mathbb{G}$, we slightly adapt the algorithm described in Mao [68] and describe it as follows:

1. Select a random prime q, k-bit long;
2. Compute: $p = 2q + 1$ and verify p is prime; if p is not prime, select q again;
3. Select two random numbers $j_1, j_2 \in_R Z_p^*$ and compute the generators: $g_1 = j_1^2\ mod\ p$ and $g_2 = j_2^2\ mod\ p$;

From the parameters (p, q, g_1, g_2), the public and private keys are computed as follows:

1. Select $x_1, x_2 \in_R Z_q$;
2. Compute: $h = g_1^{x_1} g_2^{x_2}\ mod\ p$

The M-El Gamal public parameters are (h, g_1, g_2, q, p) and the corresponding private key is (x_1, x_2).

Encryption

A message $m \in Z_q$ is encrypted as follows:

1. Select $r \in_R Z_q$
2. Compute: $g_1^r\ mod\ p$, $g_2^r\ mod\ p$, and $mh^r\ mod\ p$

The M-El Gamal ciphertext is the tuple: (g_1^r, g_2^r, mh^r)

Reencryption

By means of the M-El Gamal cryptosystem, a new ciphertext can be computed from an old one without being necessary decrypt the original ciphertext. Let (g_1^r, g_2^r, mh^r) be a ciphertext, the reencryption is performed as follows:

1. Select $s \in_R Z_q$
2. Compute: $g_1^r g_1^s\ mod\ p = g_1^{r+s}$, $g_2^r g_2^s\ mod\ p = g_2^{r+s}$, and $h^r m h^s\ mod\ p = mh^{r+s}$

The new ciphertext is the tuple: $(g_1^{r+s}, g_2^{r+s}, mh^{r+s})$

Decryption

In order to obtain the message m from the ciphertext (g_1^r, g_2^r, mh^r), compute:

$$m = mh^r/(g_1^r)^{x_1}(g_2^r)^{x_2}\ mod\ p$$

where (x_1, x_2) is the private key.

For simplicity, the description of the next protocols of this chapter employs a generic version of the El Gamal cryptosystems. In this version, for a group defined as above, the public key is $(h = g^x, g, q, p)$, for a generator g and a secret key (x). The El Gamal ciphertext is (g^r, mh^r), for $m \in Z_q$ and $r \in_R Z_q$. The plaintext is computed such that: $m = mh^r/(g^r)^x\ mod\ p$. See Mao [68] or Katz et al. [61] for details about this version of the El Gamal algorithm.

2.3.1 Threshold El Gamal Cryptosystem

The El Gamal cryptosystem described above considers a single party to generate the keys and decrypt ciphertexts. A threshold version of the El Gamal cryptosystem, however, is necessary for the voting scheme presented in Chapter 5. In this version, a number of parties n cooperatively generate the keys. The resulting private key is shared among the parties and decryption is performed (without reconstructing the original key) only if a minimal number of parties cooperate. That is, for a number n of parties, at least t of them are required to decrypt a ciphertext. The public key is unique, though.

A threshold cryptosystem is composed of a protocol for generating the keys and another for cooperatively decrypting the ciphertext [29]. We present next known solutions for these protocols concerning the El Gamal cryptosystem. The key generation protocol is due to Pedersen [76]. The version described, however, is a simplification of Pedersen's scheme given by Gennaro et al. [40]. The decryption protocol was proposed by Cramer et al. [29].

Gennaro et al. [40] showed that an adversary can influence the key generation of Pedersen's protocol. This scheme, though, remains secure in certain cases as they presented in [41]. Gennaro et al. also proposed a more secure key generation protocol in [40]. This protocol is less efficient than the Pedersen one, but it could be used as an alternative in our proposal.

The protocols presented next employ the generic version of El Gamal, but they can be used similarly in the M-El Gamal version described above. We consider that messages are broadcast to all parties through a bulletin board.

Key Generation Protocol

Let $O_1, \ldots, O_n$ be a number of parties (e.g. officials in a voting scheme), for each official O_i the Pedersen scheme for cooperatively generating a secret key (x) is described as follows:

1. O_i selects $r_i \in_R Z_q$ as his value for the secret key x and computes a random polynomial of degree $t-1$ over Z_q:
   $\alpha_i(u) = a_{i0} + a_{i1}u + \ldots + a_{i(t-1)}u^{t-1}$, where $\alpha_i(0) = a_{i0} = r_i$;

2. O_i computes $C_{ik} = g^{a_{ik}}$ mod p (for $k = 0, \ldots, t-1$) and publishes C_{ik};

3. O_i computes $s_{ij} = \alpha_i(j)$ mod q (for $j = 1, \ldots, n$) and sends it secretly to the official O_j;

4. O_j ($j \neq i$) checks each share (s_{ij}) received from all other O_i (for $i = 1, \ldots, n$). To perform this, O_j computes: $g^{s_{ij}} = \prod_{k=0}^{t-1}(C_{ik})^{j^k}$ mod p. If the equality is wrong, O_j publicly complains against O_i;

5. If O_i receives complaints from more than $t-1$ officials, he is excluded from the next steps. Otherwise, O_i publishes his secrets (s_{ij}) corresponding to each official that complained. These values are then used to verify the equation from the last step; if the check fails, O_i does not perform the next steps.

6. Let l be the number of officials that passed the last steps; O_i computes his part of the secret (x) as $x_i = \sum_{j=1}^{l} s_{ji} \ mod \ q$;

7. The public key is computed as: $h = \prod_{j=1}^{l} h_i \ mod \ p$, where $h_i = C_{i0} = g^{a_{i0}}$;

8. A public verification value $v_i = g^{x_i}$ for each (secret) share (x_i) is computed as: $v_i = \prod_{j=1}^{l} g^{s_{ij}} \ mod \ p$, for $i = 1, \ldots, l$.

Note that the shares are generated in a similar way to Shamir secret sharing scheme [90]. This scheme is based on the Lagrange interpolation of polynomials.

Decryption Protocol

We describe now the decryption scheme of Cramer et al. [29]. Let $(a, b) = (g^r, mh^r)$ be an El Gamal ciphertext, the scheme is as follows:

1. O_j computes $w_j = a^{x_j}$ using his share x_j and publishes w_j;

2. O_j proves in zero-knowledge that $log_g h_j = log_a w_j$ by using the discrete log equality test (see Section 2.5.3);

3. Taking into account that the proofs of any subset δ of t officials are correct, the plaintext is computed as: $m = \frac{b}{\prod_{j \in \delta} w_j^{\lambda_{j,\delta}}}$

   for the appropriate Lagrange coefficients: $\lambda_{j,\delta} = \prod_{l \in \delta \setminus \{j\}} \frac{l}{l-j}$

2.4 Commitment Schemes

A commitment scheme is a protocol which consists of two stages performed by two parties, a sender and a receiver. In the first stage, the sender commits to a value without revealing the value itself to the receiver. In a second stage, the sender decommits the value such that the receiver learns the value committed before.

These protocols have two fundamental properties, namely binding and hiding. The binding property means that, after being committed to a value, the sender cannot change it. The other property says that the receiver does not obtain any information about the value committed before the sender reveals it.

A simple commitment scheme is accomplished through a secure cryptographic hash function. These functions have the following usefull properties. They receive as input an arbitrary length value ($m \in \{0,1\}^*$) and output a value a of fixed length n ($a \in \{0,1\}^n$). They are also hard to invert, which means that it is difficult to find a message m from the output value a. In addition, it is hard to find two inputs which have the same output, that is, they are collisions resistant. We describe below a commitment scheme using a hash function.

1. The Sender:
    a) Selects a random number r from a large key space;
    b) Let H be a secure hash function (e.g. SHA-512 [72]) and a message m, the sender computes $H(m,r)$ as the commitment of the message m;
    c) Forwards $H(m,r)$ to the receiver;
2. a) The sender decommits $H(m,r)$ by revealing m and r later on;
    b) The receiver computes $H(m,r)$ and compares it with the commitment received before.

An introduction to commitment schemes can be found in Damgård [31] or Goldreich [42, 43]. A scheme based on discrete log can be found in Pedersen [75]. For more details about hash functions, see Menezes et al. [69].

2.5 Zero-Knowledge Proof Protocols

Zero-knowledge protocols [44, 45] are interactive proof systems between two parties: a prover and a verifier. In these protocols, the prover asserts that he knows a secret and convinces the verifier about it. The verifier, however, does not learn anything about the secret except that the assertion is true.

These protocols consist typically of three rounds and their concept is as follows: the prover and the verifier have a common input, whereas the prover has a private input (i.e. the secret about he wants to convince the verifier). The prover first commits to some information and sends it to the verifier; after this, the verifier returns him a challenge; finally, the prover sends back a response and the verifier checks whether the claim is true or false. As these proofs are probabilistic and the prover might guess the challenge, the protocol is usually ran a number of times (for new values of commit, challenge, and response) to convince the verifier with high probability. Section 2.5.1 clarifies these protocols by presenting an application of this idea.

The concept of zero-knowledge above requires the prover to interact with the verifier. That is, the prover and the verifier exchange messages in a three round protocol. Interactive zero-knowledge protocols, however, can be converted into non-interactive applying the Fiat-Shamir heuristic [35]. This way, the verifier does not need to interact with the prover to verify the assertion. To accomplish this, the prover basically uses a secure hash function to produce the challenge from his first message.

A formal definition of zero-knowledge protocols is given by Goldwasser, Micali, and Rackoff [44, 45]. See Damgård [31] or Goldreich [42, 43] for a detailed introduction to these protocols.

In the following sections we present the zero-knowledge protocols used in the voting protocol presented in Chapter 5. The descriptions are based on non-interactive versions as the voting scheme requires them to be.

2.5.1 Schnorr Signatures

The El Gamal cryptosystem is originally not plaintext-aware. That is, by means of the reencryption property, a new ciphertext may be generated from an old one without knowledge of the corresponding plaintext. This, though, is undesirable in certain steps of our proposal. In order to prevent the use of this property, we employ Schnorr signatures [87] so that the party who creates the ciphertext proves the knowledge of the random exponent and consequently of the plaintext.

Let (g^r, mh^r) be an El Gamal ciphertext of a message m, the common input is g^r as well as the El Gamal public parameters, and the prover's private input is r. The protocol is described as follows:

1. (Commit) The prover selects $t \in_R Z_q$ and computes $I = g^t \ mod \ p$. He then sends I to the verifier;
2. (Challenge) The verifier selects $c \in_R Z_q$ and sends c to the prover;
3. (Response) The prover computes $J = t + rc \ mod \ q$ and sends J to the verifier;

The verifier accepts the proof if $g^J \overset{?}{=} I(g^r)^c \ mod \ p$, that is, $g^{t+rc} \overset{?}{=} g^{t+rc}$.

In the non-interactive version, the prover uses a secure hash function to compute the challenge. The scheme now works as follows:

1. The prover:
   a) Selects $t \in_R Z_q$;
   b) Computes $I = g^t \ mod \ p$;
   c) Computes the challenge as $c = H(I, g^r) \ mod \ q$;
   d) Computes $J = t + rc \ mod \ q$;
   e) Sends (I, J, c) to the verifier.
2. The verifier checks: $g^J \overset{?}{=} I(g^r)^c \ mod \ p$

2.5.2 Proving Knowledge of a Representation

In Schnorr's scheme described above, the prover proves the knowledge of one discrete log. Okamoto [73], though, introduced a protocol where a composition of two discrete logs can be proved. That is, proving the representation of an element with regard to a set group generators[1]. By means of Okamoto's protocol, the representation of $(a^s h^r)$, for $s, r \in_R Z_q$ and two public group generators (a, h), is proved as follows:

1. The prover:

   a) Selects $t_1, t_2 \in_R Z_q$;

   b) Computes $I = a^{t_1} h^{t_2}\ mod\ p$

   c) Computes the challenge as $c = H(I, a^s h^r)\ mod\ q$;

   d) Computes $J_1 = t_1 + cs\ mod\ q$ and $J_2 = t_2 + cr\ mod\ q$;

   e) Sends (I, J_1, J_2, c) to the verifier.

2. The verifier checks: $a^{J_1} h^{J_2} \stackrel{?}{=} I(a^s h^r)^c\ mod\ p$.

2.5.3 Discrete Log Equality Test

In the discrete log equality test, the prover knows a secret r and convinces the verifier that two values g^r and a^r (where g, a are public group generators) were both generated with the same r. That is, he proves that $log_g g^r = log_a a^r$ holds for the bases g and a. This primitive is owing to Chaum and Pedersen [20]. The proof is performed as follows:

1. The prover:

   a) Selects $t \in_R Z_q$;

   b) Computes $I_1 = g^t\ mod\ p$ and $I_2 = a^t\ mod\ p$

   c) Computes the challenge as $c = H(I_1, I_2, a^r, g^r)\ mod\ q$;

   d) Computes $J = t + cr\ mod\ q$;

   e) Sends (I_1, I_2, J, c) to the verifier.

2. The verifier checks: $g^J \stackrel{?}{=} I_1(g^r)^c\ mod\ p$ and $a^J \stackrel{?}{=} I_2(a^r)^c\ mod\ p$

[1] Finding a representation of an element is known as representation problem (see Brands [13] for details).

2.5.4 Applying Okamoto's and Chaum-Pedersen's Solutions

The voting scheme presented in Chapter 5 requires a protocol for proving that a value o^s and an El Gamal encrypted value a^s have the same exponent. This proof can be accomplished using a variant of Okamoto's and Chaum-Pedersen's protocols, such as the solution presented by Redz [79]. The scheme of Redz was originally proposed to accomplish proofs for the plaintext equivalence test protocol (see Section 2.5.5) and uses Pedersen commitments [75]. But it also performs the proof that our scheme requires as described below.

Let $(g^r, a^s h^r)$ be an El Gamal ciphertext of the plaintext a^s and a value o^s, where $r, s \in_R Z_q$ and g, o, a are public generators. The protocol to prove that o^s and the encrypted a^s have the same exponent is as follows:

1. The prover:
   a) Selects $v, t \in_R Z_q$ and computes: $I_1 = o^v \ mod\ p$ and $I_2 = a^v h^t \ mod\ p$;
   b) Computes the challenge as $c = H(I_1, I_2, a^s h^r, o^s) \ mod\ q$;
   c) Computes: $J_1 = v + cs \ mod\ q$ and $J_2 = t + cr \ mod\ q$;
   d) Sends (I_1, I_2, J_1, J_2, c) to the verifier.
2. The verifier checks: $o^{J_1} \overset{?}{=} I_1(o^s)^c \ mod\ p$ and $a^{J_1}h^{J_2} \overset{?}{=} I_2(a^s h^r)^c \ mod\ p$

2.5.5 Plaintext Equivalence Test

The plaintext equivalence test is a primitive in which the plaintexts of two El Gamal ciphertexts (made with the same key and different random factors) can be verified regarding their equality. This is performed without decrypting the ciphertexts themselves or revealing any information about the plaintexts apart from if they are equal or not. We describe below a distributed version of this protocol due to Jakobsson and Juels [49].

Denote $(\alpha, \beta) = (a/c, b/d)$ be a new ciphertext created from two ciphertexts $(a, b) = (g^r, m_1 h^r)$ and $(c, d) = (g^t, m_2 h^t)$, for $r, t \in_R Z_q$ and two messages m_1, m_2. In order to check whether $(m_1 = m_2)$ or not, a set of parties first cooperatively blind the ciphertext (α, β). That is, each party raises (α, β) to a random secret value r such that $(\alpha', \beta') = (\alpha^r, \beta^r)$. The parties then decrypt the resulting ciphertext and obtain 1 if $(m_1 = m_2)$ or a random value if $(m_1 \neq m_2)$.

For a number of parties $O_1, \ldots, O_n$ and a bulletin board communication model, each party performs the protocol as follows:

1. O_i selects $r_i, t_i \in Z_q$, computes the Pedersen commitment [75] $C_i = g_1^{r_i} h_1^{t_i}$, and publishes it. For a public group generator g_1 and $h_1 = g_1^{x_1}$, where no one has knowledge of the key x_1;
2. O_i computes $\alpha_i = \alpha^{r_i}$ and $\beta_i = \beta^{r_i}$ and publishes (α_i, β_i);

3. O_i computes a zero-knowledge proof that knows the values r_i, t_i related to C_i, α_i, and β_i and publish it.

4. All parties computes $(\alpha, \beta) = (\prod_{i=1}^{n} \alpha_i, \prod_{i=1}^{n} \beta_i)$ and cooperatively decrypt (α, β) by using the decryption protocol presented in Section 2.3.1.

The zero-knowledge proof required by the scheme in step 3 can be found in Redz [79].

2.5.6 Proof of Validity of an Encrypted Vote

The voting scheme presented in Chapter 5 also requires a proof that a ciphertext encrypts a valid vote. That is, the prover convinces the verifier that the ciphertext contains a vote for one of the options available without revealing the option itself.

The protocol presented below was proposed by Lee et al. [64] (see also [63]). The prover proves that the encrypted option is 1 out k possible options. This protocol is a variant of Cramer et al.'s [29] solution. The idea behind the scheme is to prove the relation $log_g(x) = log_h(y/m_1) \ \vee \ \ldots \ \vee \ log_g(x) = log_h(y/m_k)$ by means of a witness indistinguishable proof of knowledge [34, 28].

Let $(x, y) = (g^r, m_i h^r)$ be an El Gamal ciphertext, where $m_i \in \{m_1, \ldots, m_k\}$ for k options available, the protocol is described as follows:

1. The prover:

   a) Selects $w \in_R Z_q$ and computes: $a_i = g^w \ mod \ p$ and $b_i = h^w \ mod \ p$;

   b) Selects $d_j, r_j \in_R Z_q$ and computes: $a_j = g^{r_j} x^{d_j} \ mod \ p$ and $b_j = h^{r_j} (y/m_j)^{d_j} \ mod \ p$, for $j = 1, \ldots, i-1, i+1, \ldots, k$;

   c) Computes the challenge as $c = H(a_1, b_1, \ldots, a_k, b_k) \ mod \ q$;

   d) Computes: $d_i = c - \sum_{j \neq i} d_j \ mod \ q$ and $r_i = w - r d_i \ mod \ q$

   e) Sends $(a_1, b_1, d_1, r_1, \ldots, a_k, b_k, d_k, r_k)$ to the verifier.

2. The verifier checks:

$$d_1 + \ldots + d_k \overset{?}{=} H(g^{r_1} x^{d_1}, h^{r_1} (y/m_1)^{d_1}, \ldots, g^{r_k} x^{d_k}, h^{r_k} (y/m_k)^{d_k})$$

2.6 Universally Verifiable Mixnets

A mixnet is a cryptographic mechanism first introduced by Chaum [18] that provides anonymity. It is usually composed of more than one mix server (say mixer) that sequentially shuffle a set of messages, encrypted in advance. That is, each mixer receives the set, shuffles it, and forwards the shuffled set to the next server. In order to perform the shuffle, the mixers individually apply a random permutation to the ciphertexts. In addition, they normally operate by decrypting or reencrypting them, depending on the mixnet method employed. The communication among the mixers can be accomplished through bulletin boards.

Decryption mixnets follow the original concept presented by Chaum. Every mixer has an asymmetric encryption key pair (e.g. RSA [81]). Each message is encrypted with the mixers' public keys. The resulting ciphertext is composed of encryption layers so that they can be removed sequentially as they cross the mixers. Mixers permute secretly the ciphertexts and remove the layers assigned to them by using their secret keys. A different concept is used in reencryption mixnets first proposed by Park et al. [74]. These mixnets employ just one key for encrypting the messages and require a cryptosystem with reencryption property like El Gamal (see Section 2.3). Each mixer reencrypts each ciphertext using a secret random factor and randomly permutes the ciphertexts.

Universal verifiability is an additional property present in mixnet schemes such as [71, 50]. As malicious mixers may compromise the shuffle, for instance by replacing messages, in these proposals the mixers publicly prove that they have performed the shuffle correctly. That is, each mixer proves that the messages output were permuted and that decryption or the reencryption were performed regarding the input messages. These proofs are normally accomplished using zero-knowledge proofs. Jakobsson et al. [50] employ a different approach in which mixers reveal part of the relation among input and output messages.

The scheme presented in Chapter 5 requires a universally verifiable mixnet to preserve voter privacy. We refer to [71, 50] for details of verifiable mixnets and to Adida [2] for a review of mixnets schemes.

2.7 Further Reading

We have briefly presented the cryptographic primitives necessary for our solutions. We refer to Buchmann [15], Mao [68], and Katz-Lindell [61] for more details about cryptographic techniques.

Chapter 3

An Enhanced Ballot Box

This chapter covers part of the works published in Workshop on Trustworthy Elections [5] - June/2007 and in 3rd international Conference on Electronic Voting [7] - August/2008. These are joint work with Ricardo F. Custódio and Jeroen van de Graaf, and with Peter Y. A. Ryan, respectively.

3.1 Introduction

Traditional paper-based voting has a number of threats associated to the tabulation phase. Votes, for instance, may be added, excluded, or replaced during this process. Aiming at addressing these threats, in 2001 Custódio [30] presented a new paper-based scheme in which voters sign votes. This is performed with handwritten signatures. Voters, though, do not sign their own votes, but random chosen votes. To accomplish this, the scheme employs a concept of ballot box called Farnel in which voters cast their votes and receive random selected ones that are then signed. The scheme, however, is not voter-verifiable, relies on physical signatures to achieve security, and requires trustworthy authorities.

In this chapter we introduce an enhanced Farnel ballot box. The novel concept contains extra functionalities and is a component of the voter-verifiable schemes presented in the next chapter.

This chapter is organized as follows: the next section reviews Custódio's voting scheme. After this, we introduce our concept of ballot box in Section 3.3 as well as its properties and discuss its implementation. Section 3.4 shows the process of initialization of the new box. Next, in Section 3.5, we introduce the parameters of the box and discuss its specification. Finally, this chapter is concluded in Section 3.6.

3.2 The Farnel Voting Scheme

In order to show how the Farnel box performs originally, we present here a description of Custódio's proposal.

The Farnel voting scheme uses *two* ballot boxes, that is, a Farnel box and a conventional box. The former is a special box that shuffles votes. This box is publicly initialized before the voting starts with ballots filled out and signed by a honest authority. The set of initial ballots represents, with an equal probability, all possible votes. The conventional ballot box starts the voting empty.

In order to vote, the voter receives a blank valid ballot (signed by a honest ballot authority), makes her choice, and casts the ballot into the Farnel ballot box. Then, through manual or mechanical shuffling, the Farnel box presents a ballot, chosen randomly from its current set of votes, to the voter. After receiving the ballot, the voter signs and drops it into the conventional box.

After the voting period has finished, the authority opens and signs a second time all the votes of the Farnel box and adds them into the conventional box. Then the conventional box is opened and all ballots are counted. From this result the ballots from the initialization step are discounted.

The original Farnel scheme is not voter-verifiable. That is, the voters do not obtain receipts to verify their votes were correctly tallied. However, the scheme has interesting properties. Anyone can verify that all ballots were signed by the precinct and that some of them were also signed by the voters. Therefore, ballots cannot be replaced because the fraudulent votes are not signed by the honest authority. Moreover, anyone can check who voted without obtaining the list of voters. This is performed by checking the voters' signatures on the votes. Addition and exclusion of votes after the voting phase can be detected by checking the total number of votes.

3.3 New Functionalities of the Farnel Ballot Box

In the original concept introduced by Custódio, the Farnel box is able to shuffle its contents. For this, it is supposed to have a shuffling mechanism. In addition, the box is initialized with a set of votes before the voting starts. At time of voting, after receiving a vote, it shuffles the vote along with other previous cast votes and outputs *one* random vote. The extra votes cast before the voting are subtracted from the total in the tabulation.

The voter-verifiable schemes presented in the next Chapter are based on the Farnel idea. However, in order to accomplish the schemes, we expand the original concept. In other words, we supply the Farnel box with two more functionalities. That is, besides shuffling its contents, the box is able to remove scratch surfaces and to copy some of its elements. From these additional features, we describe the new concept of the Farnel box as follows:

It is a ballot box equipped with mechanisms to remove scratch surfaces, to shuffle its elements, and to make copies in a memoryless way. The box contains an initial set of elements cast before the voting. At the time of voting, it is able to receive an element, to add it to its initial set, to shuffle the new set, to copy one or more randomly selected elements from its set, and to output the copies.

In the schemes introduced later, the elements of the enhanced Farnel box are either votes or receipts. That is, if the box is initialized with votes, then it receives votes from the voters. If it is initialized with receipts, then it receives receipts from the voters. However, the box outputs only receipts, that is, copies of its elements. Depending on the box elements, the receipts may be either in a plaintext or in an encrypted form. Plaintext receipts expose their corresponding votes during the voting. The choice of a particular voter cannot be traced, though. Encrypted receipts reveal their votes only after being decrypted in the tabulation. From now the terms (dummy) ballots or elements are used to refer to votes (or receipts) cast into the Farnel box.

Aiming at complementing the description of the new concept, we introduce next a formal specification of our Farnel box. We use the process algebra CSP (see Schneider [86] for more information) for this and specify the box as follows:

Let *Init* denote the initial set of dummy ballots with which the box is initialized. Let l denote the number of receipts to be output to each voter when they cast their votes and *ballots* the set of all possible ballots. Then the Farnel box will start in state *Farnel(Init)* and its subsequent behavior is defined recursively as:

$$Farnel_l(X) := cast?b : Ballots \rightarrow Farnel_l(X \cup \{b\}) \rightarrow \Box receipt!r : \wp_l(X \cup \{b\})$$

The notation $\wp_l(X)$ denotes the set of subsets of X of cardinality l.

In other words, the Farnel ballot box is parametrised by the integer l and its initialization *Init*. At any point, the box accepts a ballot b and adds it to its current set X. After which, it outputs a set ballots of size l chosen at random from its new set $X \cup \{b\}$.

We employ this concept in the rest of this work and assume that the Farnel box selects its elements uniformly at random to perform the copies.

An Alternative Concept

In the description above, the Farnel box shuffles the element that it receives along with its elements before outputting the copies. This way, a copy of the element received may appear in the output. However, the box may perform differently. That is, instead of adding the input element to its set first and then shuffling, the box adds the element to the set only after outputting the copies. As consequence, the input element has no chance to be copied before the next input. Considering the notation introduced above, this variation of the new Farnel concept can be specified in the process algebra CSP formally as follows:

$$Farnel_l(X) := cast?b : Ballots \rightarrow \Box receipt!r : \wp_l(X) \rightarrow Farnel_l(X \cup \{b\})$$

As before, the notation $\wp_l(X)$ denotes the set of subsets of X of cardinality l.

The Farnel ballot box now is parametrised by the integer l and its initialization *Init*. The box receives a ballot b at any time. After this, it outputs a set ballots of size l chosen at random from its current set X. Then, the new ballot is added to X and the box is ready to receive the next ballot.

3.3.1 Security Properties

The original Farnel box performs by anonymizing votes. That is, it receives a vote, shuffles it jointly with other previous cast votes, and outputs a random selected vote. This property is inherited by the enhanced box as well. However, there are two differences. First, the elements of the new box are votes or receipts. Second, it outputs copies of its elements instead of the original ones.

In order to anonymize the elements received, the original and the enhanced Farnel box depend on an initial set of dummy elements. This initial set helps preserving the voter's anonymity as will be discussed later. The anonymity afforded by the original Farnel as well as by its enhanced version, though, relies on the following requisite:

Requisite of the Farnel Box. *The dummy elements and the voters elements cannot be distinguished.*

In other words, this requisite means that no one should be able to verify the elements output by the box are dummy or not. On the contrary, the Farnel box may not ensure anonymity. Since the dummy elements are shuffled along with the voters elements, these elements help anonymizing the voters elements. However, if the distinction is possible, one needs just to distinguish the voters elements to violate the anonymity of a voter. For example, considering the new Farnel box, the first voter may vote and receive a copy of her own element as receipt; if this happens, the voter's anonymity would be violated as all other elements are dummy.

Note that the box anonymizes elements at time of voting. Thus, in principle, the contents of the box do not need to be anonymized again during the tabulation. Though, this depends on how the Farnel box is initialized as will be presented in Section 3.5.

Besides anonymizing elements, the enhanced Farnel box makes possible the verification of its contents. In order to accomplish this, the box maintains all elements it receives and outputs *only* copies of them. The copies are held by the voters as their receipts and are used to verify the original elements and votes during the tabulation.

This property of the enhanced box enables a new variant of voter-verifiability in the schemes presented in the next Chapter. That is, instead of verifying their own ballots as normally found in the literature, the voters verify subsets of the box that may or may not include their elements.

The verification provided by the new box, however, is probabilistic. That is, because the copies are made from elements chosen at random, some elements may be copied and others not.

3.3.2 Implementing the New Concept

The enhanced Farnel box has special features not existent in a conventional box. It is able to remove scratch surfaces, and to shuffle its elements as well as to copy some of them. A tombola (i.e. a raffle drum) could form the basis of an implementation of the Farnel box. This mechanism is normally used in lottery games to shuffle tickets. It is a box that contains a slit to receive items and that can be spun to shuffle its contents.

A tombola adapted to remove scratch surfaces and a copy machine would be sufficient to simulate the functionalities of the Farnel box. The tombola could be initialized with elements by officials. During the voting, after the tombola receives an element, the officials would spin it to shuffle its contents and then take one or more random elements from it. The elements then would be copied and cast back into the tombola. These procedures would be observed by voters and by helper organizations to detect possible manipulation.

A more sophisticated Farnel box, however, could be built using the current technology. The tombola could have a scanner adapted in its slit and a small printer inside it. This way, the tombola would perform all functionalities of the enhanced Farnel box without the contact of the authorities with the original elements.

Besides being implemented as a physical ballot box, most of the new concept may be employed in an electronic version. That is, with exception of the mechanism for removing scratch surfaces, the shuffling and the copies could be accomplished by a computer. For example, a direct recording electronic (DRE) voting machine may generate internally votes as dummy elements. Each time a voter votes using the DRE, it shuffles its contents and prints copies of random votes as receipts. Cryptographic mechanisms are necessary to detect or prevent malicious behaviors.

3.4 The Initialization Process

As described before, the Farnel box is initialized with dummy ballots. This process takes place before the election and is performed by the officials in a public session. The main objective is to cast a predefined number of dummy ballots (i.e. votes or receipts) into the Farnel ballot box and to publish the number of ballots cast per option on a bulletin board.

The initialization of the Farnel box is necessary mainly for ensuring the anonymity of the early voters. As the Farnel receives a ballot from each voter and outputs copies of random selected ballots, it must have an initial set of elements to choose from. Otherwise, after receiving early ballots, the Farnel would not have enough elements to select at random and copy.

For some of the schemes that we present in Chapter 4, it is necessary to ensure that ballots cast during the initialization are well-formed in some way. This will typically involve some form of random auditing. Thus, for example, we might require that $2x$ blank ballots be created beforehand. The officials perform the following steps to initialize the ballot box:

1. Select x blank ballots at random and audit them as necessary. Ballots audited are discarded;

2. Mark the other x unaudited blank ballots according to the number of votes per option specified in advance;

3. Cast the x votes (or receipts) into the Farnel box and publish the number of elements cast on the bulletin board.

Notice that some schemes in the next chapter employ a conventional and a Farnel box. In these schemes, the conventional box is initialized with votes and the Farnel is initialized with the corresponding receipts. Also, for schemes using plaintext ballots, the auditing for well-formedness is not necessary and is omitted.

In order to prevent manipulation, the initialization process should be scrutinized by helper organizations. They should check the ballot box is empty before it is initialized as well as verify all procedures above are performed correctly. Further, the ballot box should be sealed and continually supervised by third parties after the initialization. The seal is removed when the voting starts.

Initialization of the Farnel box with void ballots

Where we are using encrypted receipts we have an alternative way to initialize the Farnel box: we include a *void* option on the ballots and initialize the box with ballots representing votes for the void option. This has the advantage that we do not have to keep a log of the actual votes cast for each option during initialization. We do need a robust mechanism to ensure that all initializing votes are cast for void, but it seems likely that this is easier to enforce than maintaining a record of an initial tally. We can use this approach for the Prêt-à-Voter and ThreeBallot style ballots, for example, but not where plaintext receipts are used.

3.5 The Parameters of the Farnel box

The Farnel box is initialized with a number of dummy elements (votes or receipts) before the voting starts and outputs copies of its elements during the voting, as presented. The initial elements ensure the voter's anonymity while the copies are handed to the voter as her receipt. The number of dummy elements as well as the number of copies given to each voter compose the parameters of the box.

Because the Farnel box outputs copies of random elements of its contents, it may reveal information that affects the voter's anonymity. For example, from the copies the box could show that an element was cast before others. The information remains concealed until the tally if encrypted receipts are employed. However, it would be revealed after the receipts be decoded (unless an anonymizing mix tabulation is used). The quantity of information revealed increases according to the number of copies output by the box. The more copies the Farnel box outputs, the more information about its elements it reveals. Consequently, although more elements can be verified by the voters, the information revealed may be sufficient to compromise the anonymity of one or more voters.

In order to preserve the anonymity of the voters, the dummy elements and the voters elements cannot be distinguished through the copies output by the Farnel box as stated in Section 3.3.1. The number of dummy elements is fundamental for guaranteeing this. As the box outputs copies of previous cast elements for each voter, the elements of the early voters have more chance to be output. Hence, these elements may be distinguished from other elements. Depending on the number of dummy elements, however, the chance of distinction is negligible as the dummy elements may also be output.

To achieve verifiability while maintaining anonymity, the number of dummy elements and the number of copies should be defined according to the following requisites:

Requisite 1. *The voter's anonymity is preserved even if the Farnel box is able to output a copy of her element.*

Requisite 2. *An individual receipt or a set of them do not provide enough information to distinguish dummy elements from voters elements.*

Requisite 3. *The number of copies of elements in all receipts is sufficient to detect accuracy problems with an acceptable probability (i.e. the probability that the corruption of any given ballot is detected be at least 50%).*

We require that the voter should not be able to obtain any information other than the options order of her ballot or the option she marked when casting her ballot.

Taking into account these requisites, we consider a number of possible strategies for initializing the box: ballots marked at random (with the totals carefully recorded), a predetermined number of votes per option, votes for a void option, or a combination of these methods. If we adopt an initialization with votes for void, we must include a minimal number of votes for the other options. Otherwise, the first voter may vote and receive a copy of her own vote as receipt.

An initialization purely with void votes works only if we have mixes during the tabulation. Although the anonymity is already provided by the Farnel mechanism, it might still be useful in some contexts and does provide an extra layer of protection.

In principle, an initialization with at least one element for each option may be sufficient to preserve the voters' anonymity. However, depending on the number of voters and on the number of options, the anonymity may not be preserved. For example, considering a voting with two options, one initial element for each option, and two copies per voter as receipt. After the first voter casts her vote, she may receive her element and an initial element for the same option from the Farnel box.

3.5.1 Receipts and Verifiability

The Farnel box makes receipts for voters from elements of its contents. A receipt is a copy of a random selected element and voters may receive one or more receipts.

The probability of selection of an element depends on the number of items in the box. The more elements the box has, the less is the chance of an element to be selected. In addition, the probability is related to the number of receipts in which an element may appear. The more receipts an element may appear, the more is its chance to be selected. Elements cast early, in particular, have more chance to be selected than others as they may appear in more receipts. For a number of voters n and one receipt per voter, for example, the element of the first voter may appear in n receipts while the element of the last voter may appear only in the last receipt.

Considering the contents of the box when a voter casts her element, the probability of a copy of this element to appear at least once among the receipts is as follows. Let n denote the number of voters, *Init* denote the number of initial (dummy) elements, and l denote the number of receipts output per voter, the probability to select the element of the voter i ($1 \leqslant i \leqslant n$) is given by:

$$\Pr[\text{element } i] = 1 - \Big[\prod_{j=i}^{n}\Big(\frac{\mathit{Init} + (j-1)}{\mathit{Init} + j}\Big)^{l}\Big] \tag{3.1}$$

Note that when the first l receipts are made, the box contains the initial elements and the element of the first voter (i.e. $i = 1$). Thus, the initial elements have the same probability as the first element cast. Section 3.5.2 contains experiments for different values l and *Init*.

The receipts aim at verifying the elements of the box and should detect corruption of these elements. However, the different probabilities of the elements affects the verifiability. Elements with less chance to be output may not appear among the receipts and consequently cannot be verified. Conversely, elements with more chance are more probable to be verified.

Apart from the different chances that the elements have, some elements may not have receipts as the box performs random selections. Moreover, the box may have more elements than the number of all possible receipts. Elements without receipts could be corrupted by an adversary, but the identification of these elements is hard to perform. First an adversary needs to observe all receipts to identify these elements. The fact that some voters may refuse to do that renders this difficult. Second, depending on the parameters defined, the Farnel box is able to output receipts for most of its elements and this can be increase with extra receipts (see below).

In general, the corruption of an element takes place at random. That is, during the tabulation, an element is randomly chosen and then corrupted. The corruption should be detected by means of the receipts corresponding to the element. However, as some elements may not have had copies output by the box, the detection is probabilistic. The probability of detection of a corrupted element is as follows:

$$\begin{aligned}\Pr[\text{detect a corruption}] = &\Big(\frac{\mathit{Init}+1}{\mathit{Init}+n}\Big)\Pr[\text{element } \mathit{1}] + \\ &\sum_{j=\mathit{Init}+2}^{\mathit{Init}+n}\Big(\frac{1}{\mathit{Init}+n}\Big)\Pr[\text{element } j] \end{aligned} \tag{3.2}$$

In this equation we suppose that all voters employ their receipts to verify the corresponding elements and that corruptions take place *only* in the tallying phase. Because the box is just opened in the tabulation and the elements are protected inside it, we do not consider corruptions before this phase. In addition, the voting process is observed by everyone (i.e. voters, observers, officials, etc.). This prevents access to the box elements before the voting closes. See Section 3.5.2 for experiments with different values l and *Init*.

The equations presented above are related to the requisites of the parameters introduced before. Requisites 1 and 2 corresponds to the anonymity and are related to the Equation 3.1. Requisite 3 corresponds to the verifiability and consequently to Equation 3.2.

Increasing the Verifiability with Extra Receipts

The fact that elements cast later have less chance to have copies among the receipts affects the verifiability of the box contents. In order to increase the chance of these elements, however, the box may output a set of extra receipts at end of the voting. That is, after the last voter has obtained her receipt, the Farnel box shuffles its contents (without receiving an input), copies a number of random selected elements, and outputs them. These receipts would be handed to helper organizations and could be also published on the bulletin board.

As alternative, the box could also output additional receipts (for helper organizations) during the voting. These receipts would increase the chance of other elements to appear on the receipts.

3.5.2 Parameters Specification

The parameters of the box should be specified according to the requirements presented in Section 3.5. That is, the parameters should preserve the voters' anonymity, should detect corruption of elements with sufficient probability, and should prevent the distinction between voters and dummy elements. Because the parameters are related to the voters' anonymity and to the voter verifiability, the specification depends on the number of voters. This way, we have three variables to consider: the number of dummy elements *Init*, the number of receipts per voter l, and the number of voters n.

The value *Init* has particular concern as the Farnel box can be initialized in several ways (i.e. ballots marked at random, void ballots, etc.). Depending on the kind of initialization, more elements per option may be necessary to ensure anonymity and consequently the value *Init* increases. For now, we ignore how the box is initialized.

We investigate below the specification of the parameters with regard to the variables *Init*, l, and n. From these variables, we deduce the following:

1. A value l greater than *Init* may not ensure anonymity. The first voter, for example, may vote and obtain her element and all dummy elements as receipt.

2. For an *Init* greater than the number of voters n (e.g. 3 times greater), the dummy elements will appear in more receipts than the voters elements. As result, the elements cast by the voters and the dummy elements will be almost statistically indistinguishable. If l is small (e.g. $l = 1$), an adversary cannot violate the voters' anonymity (in particular, the anonymity of the early voters). A small l, though, affects the voter verifiability as the chance of detecting corruption of elements decreases. The anonymity may be also ensured for large values of l.

3. An *Init* equal to n also results in more dummy elements on the receipts as these elements have more chance to be selected than the voters elements. The dummy elements and the voters elements, though, are still almost indistinguishable as the number of voters elements does not exceed the number of dummy elements. The anonymity, though, can only be ensured for small values of l; a greater l may endanger the anonymity.

4. An *Init* less than n produces more voters elements on the receipts than dummy elements. Consequently, an adversary has more chance to distinguish between the voters and the dummy elements. The distinction may be possible even if l is small.

By these deductions, we observe that the verifiability and the anonymity properties are related. That is, more dummy elements mean more anonymity, but less verifiability; conversely, more receipts mean more verifiability, but less anonymity. In order to elucidate this relation, we present next the results of experiments.

The experiments employ the Equations 3.1 and 3.2 presented in Section 3.5.1. They show the chances of all voters' elements to appear at least once in the subsequent receipts as well as the chance of detecting a corruption of an element through the receipts. The experiments were performed with different values for *Init* and for l, but with a fixed number of voters $n = 500$. The value 500 is about the total number of votes cast per ballot box (or DRE) in Brazil. We employ a number of receipts l from 1 to 5. Although a greater number of receipts could be used, this would require more work from the voters to verify their receipts and consequently a number of them could ignore the verification.

- Experiment 1 - $\mathit{Init} > n$

The first results are presented in the Figures 3.1 and 3.2. These figures show the case where $\mathit{Init} > n$.

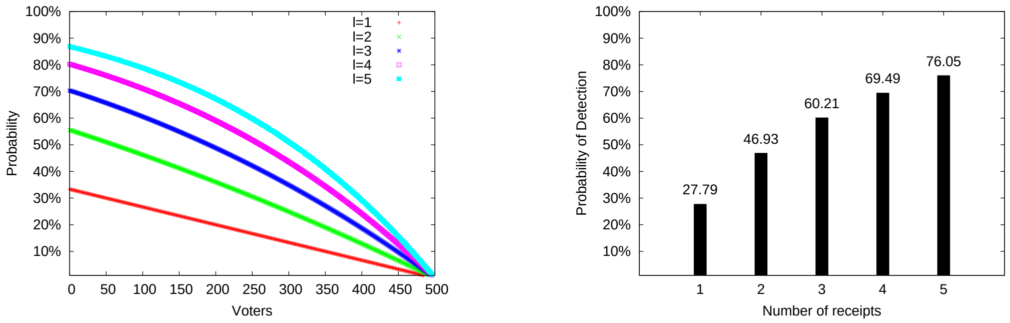
The Figures 3.1a and 3.2a show the chances of the voters' elements to appear on the receipts for $\mathit{Init} = 1000$ and for $\mathit{Init} = 800$, respectively. These results show that for $l = 1$, even the elements cast early have less than 50% of probability to have a copy output by the box. The chances, however, increase as more receipts are employed. For $l = 5$, the elements of the first 20 voters have about 85% chance in Figure 3.1a and 90% chance in Figure 3.2a. These elements as well as the elements of the next voters cannot be distinguished from the dummy elements as the *Init* value is sufficiently large and many more dummy elements will appear on the receipts. This way, the voter's anonymity is ensured.

The chances of detecting a corruption are showed in the figures 3.1b (for $\mathit{Init} = 1000$) and 3.2b (for $\mathit{Init} = 800$).

- Experiment 2 - $\mathit{Init} = n$

The experiments presented in Figure 3.3 show the case where $\mathit{Init} = n$, for $\mathit{Init} = n = 500$.

The chances of the voters' elements to appear on the receipts are presented in Figure 3.3a. Due to the reduction of the number of dummy ballots, we observe now that the chances increased compared to the last experiments. Especially, for $l = 5$, the elements cast by the first 40 voters have about 95% chance to appear on the receipts. Taking into account this number of receipt, the number of dummy



(a) Probability of the elements to appear at least once on the receipts.

(b) Probability of detecting a corruption through the receipts.

Figure 3.1: Experiments for a number of dummy elements $Init = 1000$.

elements is still sufficient to preserve the voter's anonymity. Although the value $Init$ is less than in the last experiments, the receipts will still be composed of many more dummy elements than voters' elements.

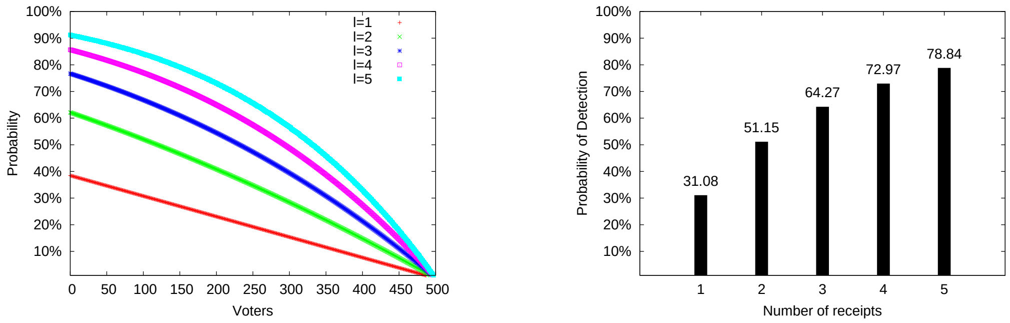
Figure 3.3b present the chance of detecting corruptions for a different number of receipts.

- Experiment 3 - $Init < n$

The last experiments show the case where $Init < n$. These results are presented in the Figures 3.4 and 3.5.

The Figures 3.4a and 3.5a show the probabilities of the voters' elements, for $Init = 300$ and for $Init = 100$. We call attention to the case where $l = 5$ in both figures. In these results, for $Init = 300$ the first 20 elements cast by the voters appear with about 99% probability on the receipts while for $Init = 100$ this probability occurs for the first 138 elements. In the latter result, due to the high number of early elements that will certainly appear on the receipts, the dummy elements provide less anonymity for the early voters than in the experiments presented before. The reason for this is explained as follows.

Observe that the total number of receipts is 2500 (i.e. $l \cdot n$) and that the dummy elements have the same probability as the element of the first voter (i.e. about 99% here). As the dummy elements may appear in all receipts, we expect more receipts for these elements. In the same way, we expect more receipts for the first 138 voters' elements as they also have about 99% chance and may appear in more receipts than the subsequent elements. Considering all receipts and $Init = 100$, 238 elements may appear on the receipts more than others. Some of these elements are dummy and others are the elements of the first 138 voters. Consequently, the distinction between voters and dummy elements could be performed among less elements than in the simulations presented before.



(a) Probability of the elements to appear at least once on the receipts.

(b) Probability of detecting a corruption through the receipts.

Figure 3.2: Experiments for a number of dummy elements $Init = 800$.

Although when $Init < n$ more voters' elements may appear on the receipts, this fact does not mean that the anonymity of a voter may be violated. As the dummy elements appear on the receipts with the same probability of the first element cast, they anonymize the voters elements and make the distinction difficult. This anonymization, however, depends on the $Init$ value. In addition, each element cast by the voters help anonymizing the elements of the next voters.

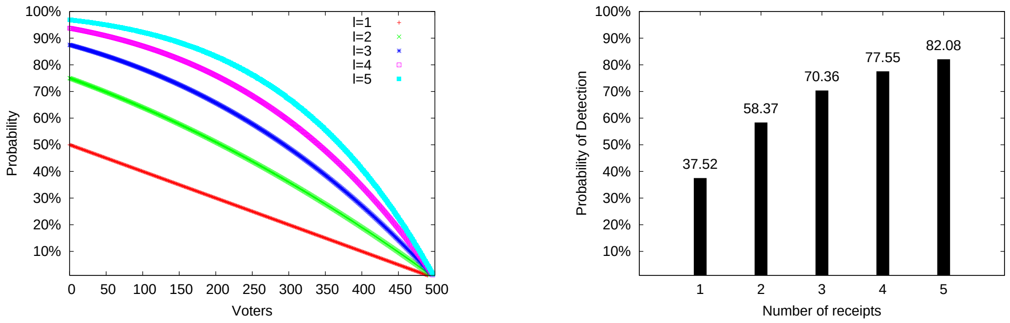
The chance of detecting a corruption through the receipts is presented in the figures 3.4b (for $Init = 300$) and 3.5b (for $Init = 100$).

3.5.3 Simulations

The last section presented experiments to clarify the anonymity and the verifiability properties. These experiments, though, were based on the equations defined in Section 3.5.1. We present now the result of voting simulations performed with different parameters. These simulations, in particular, have the goal of verifying the anonymity of the first voter could be compromised and corrupted elements can be detected.

Anonymity

We have argued in the previous section that the anonymity and the verifiability properties of the Farnel box are related and have demonstrated this through experiments. As observed, initializations with large $Init$ values (i.e. $Init > n$ or $Init = n$) provides better anonymity. These initializations, however, are only of theoretical interest if a large number of voters is considered. They would require much work from the officials and more control over the elements to decrease the risk of manipulation. In order to avoid these drawbacks, the value $Init$ should be



(a) Probability of the elements to appear at least once on the receipts.

(b) Probability of detecting a corruption through the receipts.

Figure 3.3: Experiments for a number of dummy elements $Init = 500$.

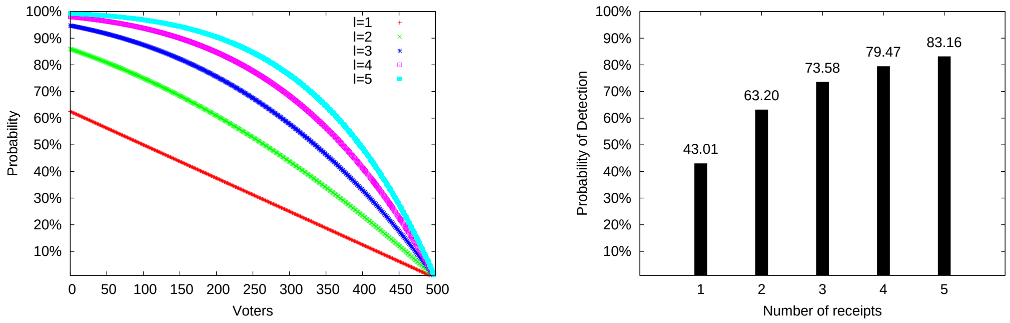
as less as possible. A value $Init < n$, hence, is more appropriate.

On the other hand, an initialization with a small *Init* value exposes more elements of early voters, as presented above. For $Init = 100$, for instance, we have showed that the elements of the first 138 voters may appear more often on the receipts than the elements of the next voters. As consequence, the receipts could reveal information about the elements of early voters sufficient to violate their anonymity. Especially, the anonymity of the first voter.

Recall from Section 3.5.1 that the element of the first voter and the dummy elements have the same probability to appear on the receipts and that these elements may appear in more receipts than others. By observing most of the receipts, one could identify the element that appeared most often and conclude that it is from the first voter. In order to verify the voter anonymity can be violated in this way, we have implemented a simple program to perform simulated voting.

The program implements the Farnel box by means of a linked list [27]. It considers two candidates and initializations with the same number of elements for these candidates. The elements are votes that are randomly generated and are added to the list in random positions; each vote is identified uniquely. The receipts are copies of randomly selected votes (identifiers) from the list. For each election, after sorting the list based on the number of receipts, the program returns the number of receipts in which the vote of the first voter appeared on and the position of this element in the sorted list. Also, it shows whether more votes had the same number of receipts as the vote of the first voter.

The simulations were performed with values $Init = 10, 20,$ and 30. For each of these values, 20 voting were performed. The number of dummy votes for each candidate employed was half of *Init*. Also, they considered 5 receipts per voter and 500 voters. Figures 3.6, 3.7, and 3.8 show the simulation results. Each point



(a) Probability of the elements to appear at least once on the receipts.

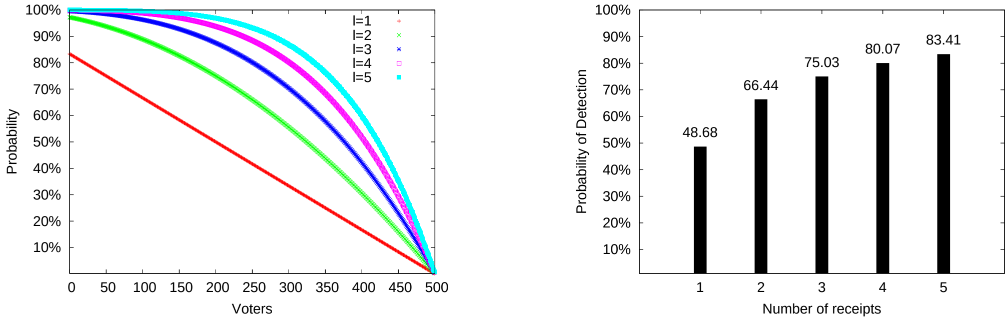
(b) Probability of detecting a corruption through the receipts.

Figure 3.4: Experiments for a number of dummy elements $Init = 300$.

on the figures means one voting. The axes y show the position of the vote of the first voter regarding the number of receipts issued in the voting. The values near the points indicate the numbers of votes with the same number of receipts as the vote of the first voter. The axes x show the number of receipts corresponding to the vote of the first voter (or others as long as the values near the points are greater than 1). In order to verify the vote of the first voter appeared more than others on the receipts, we observed if this vote is on the first position and if it is a unique element in this position.

Figure 3.6 presents simulations for $Init = 10$. In these simulations we expect 11 votes to appear more on the receipts (i.e. ten dummy votes and the vote of the first voter). The simulations show that the anonymity of the first voter would be violated in 2 out of 20 voting. That is, the vote of the first voter appears in position 1 for a number of 26 and 28 receipts. Note that for 26 receipts another vote obtains the same number of receipts as the voter's vote. In this case, if the votes were for different candidates, an adversary would need to distinguish between the vote of the first voter from the other vote. However, considering all simulations and the range $[1; 28]$ in which the receipts corresponding to the voter's vote appear on, an adversary cannot be sure if the vote on the first position is from the first voter or not.

The voting simulations for $Init = 20$ are showed in Figure 3.7. In these simulations we expect more receipts for the 20 dummy votes and for the vote of the first voter. No voting, however, had the vote of the first voter as the one with more receipts. Comparing to the last simulations, observe that the number of receipts obtained by the voter's vote here are distributed over a larger range (i.e. $[5; 72]$); also, the number of votes that have the same number of receipts as the voter's vote has increased. This would render even more difficult the association of a number of receipts to the vote of the first vote, in a certain voting.



(a) Probability of the elements to appear at least once on the receipts.

(b) Probability of detecting a corruption through the receipts.

Figure 3.5: Experiments for a number of dummy elements $Init = 100$.

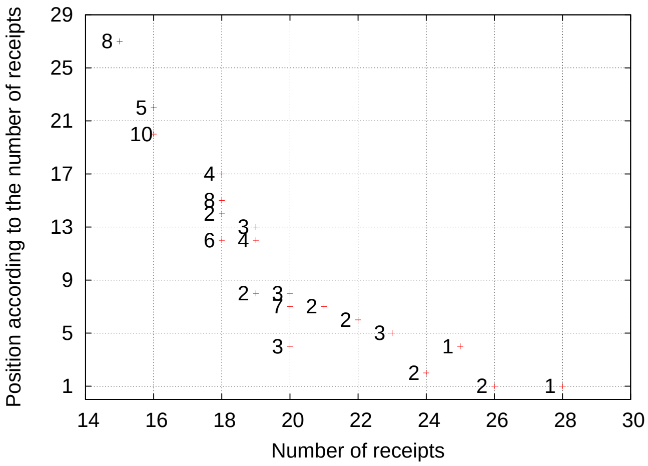


Figure 3.6: Simulations for $Init = 10$.

The last simulations are presented in Figure 3.8. They consider $Init = 30$ and the first 31 votes are the most expected on the receipts. The voting that shows 28 receipts would reveal the vote of the first voter. In the other voting, the receipts corresponding to the vote of the first voter are distributed over the range $[2; 102]$. As before, the large interval as well as the number of votes with the same number of receipts as the voter's vote impede one to violate the voter's anonymity through the receipts.

Overall the anonymity of the first voter would not be compromised in the simulations if one is able to observe all receipts. This is true since the exact number of receipts corresponding to the voter's vote cannot be determined. However, the simulations show that by reducing the number of dummy votes, the range in which the voter's vote may appear is reduced as well. If the range is much

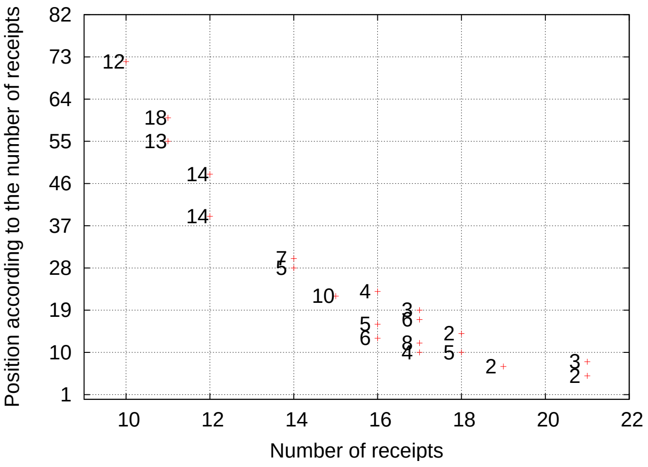


Figure 3.7: Simulations for *Init* = 20.

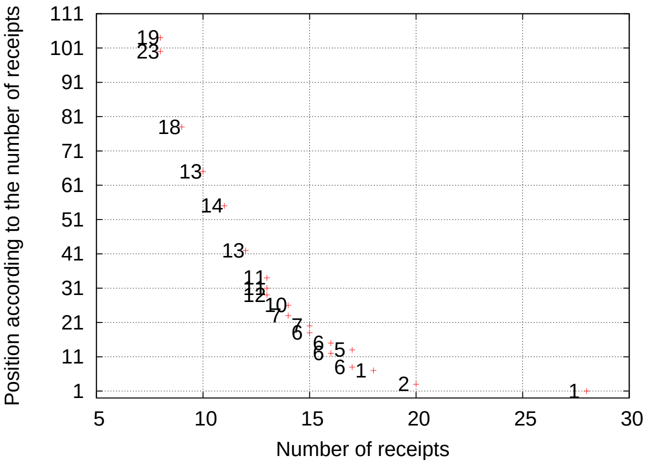


Figure 3.8: Simulations for *Init* = 30.

smaller, we deduce that the vote of the first voter would appear more often in the first position and this would endanger the voter's anonymity.

Verifiability

As stated in Sections 3.3.1 and 3.5.1, the new Farnel concept provides probabilistic verifiability since some votes may have receipts and others not. Although an adversary cannot identify votes without receipts and corrupt them, he may try to corrupt random votes after the voting.

While performing the simulated voting presented above, we have also verified whether corruptions could be detected by means of the receipts. That is, at the end of each voting, some votes are corrupted and it is verified they have corresponding receipts. In order to accomplish this, all receipts (i.e. copies of votes) generated during the voting were stored; then some votes were selected at random from the linked list (i.e. the Farnel box) and compared with the receipts. Detection of corrupted votes, hence, resulted on votes from the linked list that matched receipts.

In each voting, a number of 20 votes were corrupted at random. Because these simulations were performed along with the last ones, they have employed the same set up. That is, 500 voters, 5 receipts per voter, and a number of dummy votes $\mathit{Init} = 10, 20, \text{and } 30$. Tables 3.1, 3.2, and 3.1 show the results of 15 simulations for $\mathit{Init} = 10, 20, \text{and } 30$, respectively. Each line in the tables corresponds to a voting. The columns present the numbers of votes that were detected by the receipts and that were corrupted.

Detected	Corrupted
18	2
15	5
15	5
17	3
16	4
15	5
18	2
18	2
18	2
17	2
16	4
19	1
17	3
18	2
19	1

Table 3.1: Detecting corruptions for $\mathit{Init} = 10$.

Detected	Corrupted
14	6
18	2
15	5
14	6
18	2
18	2
16	4
17	3
14	6
16	4
17	3
17	3
19	1
17	3
16	4

Table 3.2: Detecting corruptions for $\mathit{Init} = 20$.

Detected	Corrupted
20	0
14	6
17	3
18	2
17	3
16	4
16	4
16	4
16	4
17	3
18	2
19	1
18	2
17	3
17	3

Table 3.3: Detecting corruptions for $\mathit{Init} = 30$.

As expected most of the corrupted votes would be detected for the parameters used. Though, we can observe that more dummy votes means more chance for the adversary to succeed.

3.6 Conclusions

In this chapter we have introduced a novel concept of ballot box based on the Farnel approach. In the new concept, the ballot box not only anonymizes its elements, but it also provides copies of its elements as receipts to the voters. This allows a new variant of voter-verifiability in the schemes presented in the next chapter. In addition, we have detailed the initialization process required by our proposal and defined its parameters.

We have also showed the specification of the parameters through experiments and simulations. As stated, less initial elements increase the verifiability, but may undermine the anonymity depending on the number of receipts and the number of voters. In our simulations, we have employed a small number of initial votes and verified that the privacy of the first voter was not violated. We have also performed simulations to verify votes corrupted after the voting could be detected through the receipts.

In the next chapter we assume the existence of ballot box presented in order to introduce the novel voter-verifiable schemes.

Chapter 4

Voter-Verifiable Voting Schemes

This chapter is a combination of two works: the first one is joint work with Ricardo F. Custódio and Jeroen van de Graaf. It has been published in Workshop on Trustworthy Elections [5] - June/2007 and will appear on Towards Trustworthy Election Systems book [21]. The second one is joint work with Peter Y. A. Ryan and has been published in 3rd international Conference on Electronic Voting [7] - August/2008.

4.1 Introduction

Voter-verifiable voting schemes allow voters to verify their votes are accurately counted by means of protected receipts. Because voters are able to detect counting errors or possible manipulation of the results, these schemes usually afford more security guarantees and require less trust in their components than traditional voting. Voters, though, cannot use their receipts to compromise their privacy, even if they are prepared to cooperate with adversaries.

Schemes that provide this security feature, such as Prêt-à-Voter [83, 22], usually employ cryptographic mechanisms (e.g. mixnets) to ensure security. However, Randell-Ryan [78] and Rivest [80] introduced solutions that do not require these techniques. Whereas these schemes are simple and can be more easily understood by the voters, they provide less security guarantees than the cryptographic based ones. In the proposal of Rivest, the ballot secrecy is not perfect and it may leak statistical indications about the results in the course of the voting. The scheme proposed in [78] requires extra mechanisms to prevent malicious officials from altering votes in the tabulation.

In this chapter we first introduce a novel voter-verifiable voting scheme. The proposal does not require cryptography, while avoiding receipts from leaking results before the voting ends. As the previous proposals, however, it does not achieve the same security levels as cryptography based schemes. Thus, with the goal of achieving more guarantees than our first solution, we present two schemes that employ cryptography. These schemes, though, require just a commitment protocol. In addition to these proposals, we introduce new versions of Rivest and Randell-Ryan voting schemes that aim at overcoming their drawbacks. The new solutions are based on our initial schemes.

Our solutions are paper-based and rely on the enhanced Farnel ballot box introduced in Chapter 3. Also, they employ a novel variant of voter-verifiability. That is, instead of verifying their own votes, voters are able to verify one or more random votes.

The next section presents basic aspects related to our solutions. Section 4.3 introduces a voter-verifiable scheme that does not rely on cryptography. Section 4.4 presents an improved version of the previous scheme that employs cryptography. Section 4.5 shows a scheme that requires only a Farnel box. After that, Section 4.6 and Section 4.7 present improved versions of the Threeballot and the Randell-Ryan schemes. Finally, this chapter is concluded in Section 4.8.

4.1.1 Related Work

The proposals presented in this chapter are related to Prêt-à-Voter [83, 22] and Punchscan [77] as well as their variants, such as Ryan-Schneider [85], Scratch-and-Vote [3], van de Graaf [93], and Xia et al. [96, 97] schemes. In addition, they are related to the ThreeBallot [80, 82] and to the Randell-Ryan [78] schemes. These proposals employ paper ballot and are voter-verifiable. Our proposals, however, differ from these previous works basically in two ways. First, we employ a special ballot box and anonymize votes (or receipts) at the time of voting. Instead, the ThreeBallot and the Randell-Ryan schemes require conventional ballot boxes. Prêt-à-Voter and Punchscan employ cryptographic techniques and anonymize votes during the tallying. Second, in these previous works the voter verifies her own vote was correctly tallied, whereas in our proposals she verifies one or more random votes. Because our proposals are paper based, we do not consider here Direct Recording Electronic (DRE) based schemes, such as Bingo Voting [10], Moran-Naor [70], and Chaum [19] solutions.

A solution similar to the scheme we introduced in [5] (see Section 4.3), that also relates to our other solutions, appeared independently in [82]. This scheme is also based on the original Farnel, but it employs a different concept. That is, it does not require initialization of a ballot box, it does not issue receipts for all voters, and each voter receives just one receipt.

4.2 Common Properties of the Protocols

The schemes presented later share some common properties as participants and security goals, and are related to some known attacks. We introduce here these common features regarding our solutions.

4.2.1 Participants and their Role in the Voting Process

A voting scheme involves participants that perform determined functions. We introduce next the participants and their roles in our proposals.

Voter The voter casts her vote and receives one or more receipts for checking votes later. In addition, she has the possibility to audit ballot forms in public before voting. When necessary, helper organizations assist the voter to perform this inspection. She also employs her receipts to verify votes on the bulletin board.

Voting Authorities These authorities are responsible to initialize publicly ballot boxes (e.g. the Farnel box) and audit ballot forms. In addition, they hold a set of sealed envelopes containing blank ballots and a list of eligible voters. This list is employed to authenticate voters during the voting as conventionally. Note that these tasks can also be performed by different authorities. That is, a set of authorities initializes the box and another holds the envelopes.

Talliers The talliers are responsible for opening the ballot box and for counting the votes. Here, they should also publish the contents of the ballot box on the bulletin board and inspect votes when necessary.

Helper Organizations and Observers These parties supervise the whole voting process and verify the information publicly available. Helper organizations, especially, support voters to audit ballots. This process requires computers in some schemes presented later. However, as voters are not supposed to employ their own computers to inspect ballots, they select organizations to publicly perform this. Note that all other participants may perform as observers.

We assume that the eligible voters were previously registered by one or more trustyworthy registrars.

4.2.2 Known Attacks on Voter-verifiable Schemes

Voter-verifiable schemes provide mechanisms to detect a number of threats intrinsic in traditional voting. However, some threats exist for these schemes. We describe here three of them and their countermeasures. We concentrate on these three attacks because they are strongly associated to our proposed schemes.

Chain Voting Attack

The chain voting is a threat inherent in almost all paper-based voting. In this attack, an adversary smuggles a valid blank ballot, marks an option on it, and corrupts a voter to use this ballot. After voting, the voter returns to the adversary the blank ballot received from the officials. The adversary then can use the fresh ballot to corrupt another voter in the same way. (See also Jones [56]).

A countermeasure for this attack was proposed by Harris [47]. He suggested to include on each ballot a tear-off slip containing a unique identifier. This number is recorded by the officials before handing the ballot to the voter. Before the voter casts her vote, the officials verify the slip is intact and the number on it is the same recorded, and remove the slip.

Because the schemes presented in this chapter can be adapted to include Harris' countermeasure, we disregard this attack.

Randomization Attack

In 2000, Schoenmakers [88] pointed out that the scheme proposed by Hirt and Sako [48] is vulnerable to the randomization attack. In this attack, an adversary forces the voter to cast an arbitrary vote with the goal of nullifying the voter's choice with high probability.

The randomization attack, however, is also applicable to some voter-verifiable schemes of which receipts rely on randomization. In order to perform the attack, an adversary instructs the voter to generate a receipt that has a certain property. For example, a receipt marked in a determined position. The attacker will not know what vote will be encoded, this is effectively random. The effect of this attack then is to force voters to vote for a random candidate, so nullifying their right to vote freely.

The Prêt-à-Voter and the Punch Scan voter-verifiable schemes are vulnerable to this attack. In these schemes, the voter receipt contains the position chosen by the voter. This way, an adversary may ask the voter to place her X in a specific position and to show him afterwards the receipt marked in this position.

The Farnel idea mitigates this attack as we point out. The voter can be required to replace her receipt by another one before leaving the precinct. This way, an adversary cannot determine whether the voter followed his instructions because the Farnel box returns a random selected receipt. As requisite, though, the voter's receipt cannot be exposed before being exchanged. Otherwise, an adversary could obtain information to mount the attack by observing the original receipt, for instance, the voter's selections as in Prêt-à-Voter or Punch Scan.

Most of the schemes introduced in this chapter employ a Farnel box and thus are able to overcome randomization attacks. They, however, relax the requisite of hiding the receipts before exchanging it as this can be accomplished by employing envelopes. That is, receipts are first inserted into envelopes before being exchanged. Note that the envelopes should be employed along with Harris' solution; otherwise, malicious voters could replace other receipts instead of their own.

Psychological Attacks

In voter-verifiable schemes, such as Prêt-à-Voter, the voter's receipt includes a ciphertext. This cryptographic material along with other information prevent the voter from using her receipt to reveal her vote. At same time, these information allow the voter to verify her vote. For example, Prêt-à-Voter's receipt includes an encryption and a mark for the chosen candidate (see also Section 4.4.1). The voter verifies the same encryption and mark (in the same position as the receipt) are published on the bulletin board.

The ciphertext reveals the voter's choice if decrypted from the receipt. However, because an adversary has no access to the secret key, he cannot decrypt the ciphertext and compromise the vote's secrecy. This secret key is normally shared among a set of talliers and decryptions are performed in cooperation. The adversary, though, could persuade a voter that the secrecy of her vote is not guaranteed. For example, he could convince his victim that he can extract her choice from the ciphertext and thus induce her to vote for his candidates. This attack is absolutely psychological as the adversary cannot obtain the vote from the ciphertext.

The main point in psychological attacks is that the voter's receipt corresponds to the voter's vote. This way, an adversary can use this fact and persuade voters. The Farnel mechanism, though, mitigates these attacks. Because the Farnel may exchange receipts, voters do not retain their own receipts. Thus, any fear that the vote can be extracted should be mitigated.

Observe that these attacks may work in schemes that do not employ cryptography as well. In the Threeballot scheme (see Section 4.6.1 for an overview), for example, an adversary could convince a voter that he knows the other two parts that compose the three ballots.

4.3 A Two Boxes Scheme

We present now a two boxes scheme that is voter-verifiable. It applies the new variant of voter-verifiability in which voters verify random votes. The verification may include their votes or not.

The scheme *does not employ cryptography* as the previous proposals of Rivest [80, 82] and of Randell-Ryan [78]. It relies on numbers to identify the ballots and the voter retains copies of these numbers as receipt.

4.3.1 Requisites

The Ballot Form

The scheme employs a ballot form composed of two halves. The first half is not much different from the layout traditionally used in elections. It contains a list of voting options (including a blank option) where next to each option there is a bubble to select it. It also contains an identification number (ID) which identifies the ballot uniquely and associates it to the voting. The second half contains only the same *ID*. The halves are separated by a perforation to allow detachment and the *IDs* are covered by scratch surfaces. These surfaces prevent anyone from learning the ballot *ID* before the ballot be marked.

The ballot form can be additionally described as follows. Let C be a set of options available, I a set of positive integers sufficiently large, and ID a unique number in I. The first half contains C, ID. The second half contains only the same *ID*. Figure 4.1 illustrates this ballot form.

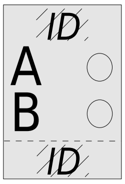


Figure 4.1: The ballot form of the two boxes scheme.

The *ID*s on both halves should be easy to compare and difficult to remember. The voter should compare the *ID*s to detect a possible malformation of her ballot form (i.e. a form with different *ID*s) and should not be able to remember it afterwards. Although these properties seem to be contradictory and difficult to implement, barcodes could be used to encode *ID*s and prevent the voters to recall them. Voters could compare barcodes easily as long as they are thick enough.

The Ballot Boxes

The scheme requires two ballot boxes. One of them is a conventional box. This box is initialized with filled out, dummy ballots (i.e. just the first half of the ballots). The other ballot box is a Farnel box as defined in Section 3.3. It is initialized with the halves of the dummy ballots that contain the *ID*s.

4.3.2 The Scheme

Before the Voting

In this phase, the authorities establish the Farnel parameters. That is, they define the number of receipts l and the number of dummy ballots *Init*. After defining *Init*, the authorities determine the number of ballots marked per candidate (see Section 3.5 for details). Also, they publicly initialize the boxes through the approach described in Section 3.4. Before this initialization, though, the authorities audit some ballots. This audit is performed by detaching the scratch surfaces and then by comparing the *ID*s of random chosen blank ballots apart.

In order to initialize the boxes, the authorities hold a number of (entire) blank ballots and tear each of them in two along the perforation. Next, they mark an option on each of the parts containing the options, detach their layers, and cast them into the conventional ballot box; the number of votes marked per option were defined before. The authorities then scratch away the layers of the other parts (the slips that contain copies of the *ID*s) and cast them into the Farnel ballot box. Note that neither the authority nor third parties should be able to record or remember the *ID*s of the dummy votes.

Voting

After proving her eligibility to the voting authorities, the voter receives a blank ballot form in a sealed envelope. She can audit this form or use it to vote. In order to audit a ballot, she detaches its scratch surfaces and compares the *ID*s in the presence of the authorities. This ballot form is discarded and the authorities hand the voter a new blank form. If any ballot fails the audit checks, then recovery mechanisms will need to be invoked. The voter performs the following steps to vote and to obtain her receipt (see also Figure 4.2).

1. **Verifying and filling out the ballot form**

   In the voting booth, the voter scratches away the layer covering the *ID*s of her ballot and matches them (a). If they are equal, she marks one of the options available (b);

2. **Casting the vote**

   The voter separates the two parts of the ballot form (c). She then casts the part containing the ballot *ID* and the options into the conventional ballot box (d). The other part, showing only the *ID*, is cast into the Farnel ballot box (e).

3. **Obtaining the receipt**

   The Farnel ballot box shuffles its contents (f) and copies l *ID*s as receipt to the voter (g).

Observe that here the Farnel box does not remove scratch surfaces and that the voter compares the *IDs*. An alternative to avoid this comparison is to use a mechanism to remove the surfaces in the box. The scheme now would require an auditing to check ballots before the voter receives her blank ballot. The voter would cast her vote without detaching the scratch surfaces.

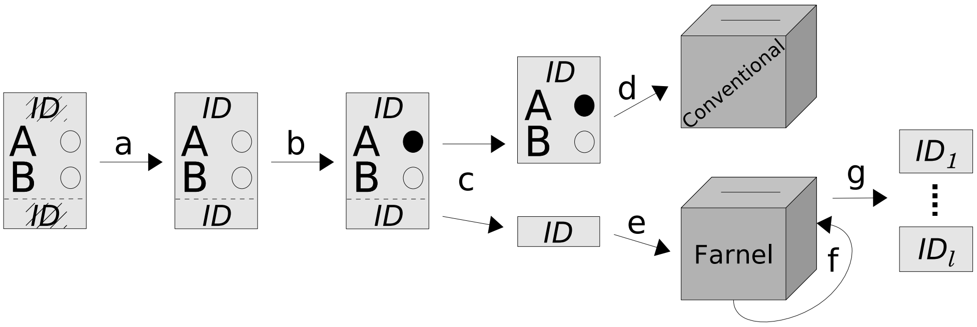


Figure 4.2: Main voting steps of the two boxes voter-verifiable scheme.

Tallying the Votes

In a public session, the talliers open the two ballot boxes and publish their contents on the bulletin board. To compute the results of the election, the talliers count all votes. The dummy votes cast in the initialization phase are subtracted from the sums yielding the final result.

Vote Verification

Anyone can check on the bulletin board whether each vote from the conventional ballot box has a corresponding *ID* in the Farnel ballot box. In addition, the voters confirm whether their receipts (i.e. the *ID*s) match to votes on the bulletin board. If one vote and its *ID* were not published, the voter complains by showing her receipt to a voting authority.

4.3.3 Evaluation

The scheme relies essentially on the special ballot form and on the Farnel box. The combination of these components along with the bulletin board preserve the voter privacy and achieve verifiability.

Upon receiving the *ID* from the voter, the Farnel box returns her copies of random selected *ID*s. As the voter cannot remember the *ID* of her own vote and as the *ID*s are random, the voter cannot violate her privacy and prove to an adversary which *ID* belongs to her vote. She can only attempt to guess the *ID* of a specific candidate from her receipts. This way, the scheme ensures privacy.

The scheme can be verified by voters and third parties. Everyone can detect duplication, elimination, substitution, and addition of votes. The detection is accomplished by checking the information published on the bulletin board. Duplicates can be identified by checking if the *ID*s of the votes published are unique. Anyone can also detect elimination and substitution of votes. Every vote on the board should have a corresponding *ID* published. The addition of votes can be detected through the total number of votes published. The total should be the sum of the number of initial votes and of the number of voters that cast their votes.

Moreover, voters can independently match their receipts (i.e. the *ID*s) to the votes on the board. Note, though, that the detection is probabilistic since not all votes may have their *ID*s printed on the receipts.

In order to achieve these security guarantees, however, the scheme requires trustworthy talliers. Also, the talliers should supervise strictly the votes from the opening of the conventional ballot box until the publication of all votes on the bulletin board. Otherwise, an adversary (e.g. a malicious tallier) may replace votes without being detected.

Suppose an adversary has access to the set of votes of the conventional ballot box before the votes are published on the bulletin board. In order to replace a vote, the adversary smuggles a vote from the set, makes a fake vote for a different option but using the same *ID* of the vote smuggled, and includes the fake vote in the set. This way, after publishing all ballots, the fake vote would appear on the board instead of the smuggled one.

This attack would undermine the accuracy of the scheme. Because votes are verified through the *ID*s, voters and third parties would not detect the replacement of the original vote by the fake one.

Hence, in order to ensure security, the scheme requires more control over the votes or other anti-counterfeiting measures to prevent the threat. We introduce next a scheme that does not need such requisites.

4.4 A Two Boxes Scheme with Cryptography

In the last section we have introduced a scheme that does not employ cryptography and that is voter-verifiable. The scheme, however, requires trustworthy talliers to fulfill this requirement. With the goal of avoiding this strong requisite, we introduce here an improved scheme based on the previous proposal.

The improved solution employs a new ballot design and relies on cryptography to achieve security.

4.4.1 New Receipt Style

The receipt employed in the last scheme has no dependence to the voter's choice. It is only connected to the options' list. This way, an adversary can replace votes without being detected. In order to detect this problem, a receipt should include some information related to the choice.

In principle, as the Farnel box anonymizes receipts, the information on the receipt could include the option chosen. For example, considering the previous scheme, we could employ the option's half as receipt instead of the *ID* half.

However, because each receipt would expose a vote, a set of receipt could reveal indications of the results before voting ends. Premature results in voting are undesirable as they could influence voters to vote for the winning candidate or the losing one.

A related problem occurs in the Twin [82] and in the Threeballot [80, 82] voting schemes. A receipt in Twin is a copy of a vote while in Threeballot a receipt reveals part of the options chosen. In both schemes, by observing a set of receipts an adversary is able to obtain statistical information about the voting results before the voting closes. Section 4.6.1 presents more details about this problem in the Threeballot scheme.

In order to overcome these problems while avoiding trust in the officials, a receipt should fulfill the following requisites:

1. It should detect accuracy problems (e.g. deletion, replacement, etc.) considering any adversary including dishonest talliers;

2. It should be related to the voter's choice;

3. It should not reveal the choice before the voting closes.

Prêt-à-Voter Ballot Design

Prêt-à-Voter [83, 22] is a cryptographic voter-verifiable voting scheme that employs a special ballot design. The ballot is composed of two halves that are separated vertically by perforations. The left half contains the options list in a randomized order. The right half contains the corresponding spaces to select the options and a mixnet onion (i.e. a ciphertext formed of encryption layers as described in Section 2.6) on the bottom. The voter marks her option on the right half and discards the left half. The receipt is a copy of the right half and the original half is cast. All the right halves are published on the bulletin board after the voting and the votes are computed from the marks and from the mixnet onion on the halves.

The ballot design employed in the Prêt-à-Voter scheme satisfies the requisites above. The receipt includes the voter's choice without revealing the choice directly. That is, as the options are randomized and the receipt contains only a mark in a determined position, the receipt does not reveal the choice. Also, because the receipt is a copy of the left half that is published on the bulletin board and the vote is reconstructed from this half, the receipt detects accuracy problems.

The schemes we introduce here and in the next sections employ ballot forms based on the Prêt-à-Voter design. They, however, do not rely on mixnet onions. The new ballot forms are also inspired by the ideas of Randell-Ryan [78] and of Scratch-and-Vote [3] schemes.

4.4.2 Requisites

The Ballot Form

The ballot form here is formed of two pages that are overlaid initially. The top page has a list of voting options in a random order and each option is associated to a bubble to select it. The top page also contains a commitment to the list of options and its respective decommitment value. The bottom page contains the *same* bubbles and the *same* commitment of the top page. The commitment printed on both pages as well as the value to open it in the top page are covered by scratch surfaces. These surfaces now prevent anyone (including the voter) from learning the commitments and its decommitment before the vote is cast. A carbon mechanism transfers the selections made on the top page to the bottom page (see Figure 4.3 for an example of this ballot form).

This design is also described as follows. Let C be a set of options available, π_C the permutation of C, H a secure hash function used here as commitment, and r a random number from a large (key) space. π_C, $H(\pi_C, r)$, r, and bubbles to select an option compose the top page. The bottom page contains only $H(\pi_C, r)$ and the bubbles in the same position as the top page.

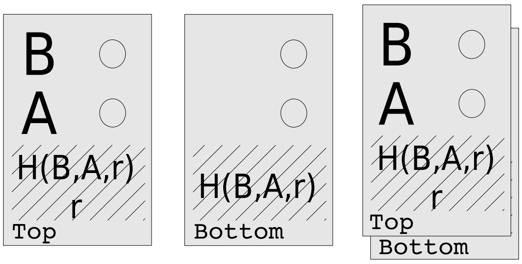


Figure 4.3: A ballot form in the two boxes scheme with cryptography. On the right the two pages of the ballot and on the left the pages overlaid can be seen.

The Ballot Boxes

As in the previous solution, the scheme employs a conventional ballot box as well as a Farnel box (see Chapter 3 for details).

4.4.3 The Scheme

Before the Voting

The authorities perform the same steps described in Section 4.3.2. That is, they define the number of receipts l and the number of dummy ballots *Init*, audit a number of blank ballots, and initialize the boxes as presented in Section 3.4.

Due to the new ballot form, however, the audit process is executed differently. In order to audit a blank ballot, the authorities now scratch off the layers on the top and on the bottom pages. After this, they hash the options along with the key r revealed on the top page. Then, they match the resulting hash with the hash values printed on the top and on the bottom page.

The conventional box here is initialized with marked top pages while the Farnel box receives the corresponding bottom pages. The authorities, though, do not scratch off the surfaces before casting the ballot pages. Upon receiving the bottom pages, the Farnel box removes their scratches; the other scratches are detached in the tallying phase.

Voting

In the voting phase, upon proving her eligibility to the officials, the voter receives a sealed envelope with a blank ballot. If required by the voter, her ballot can be audited (as above) and she receives a new blank ballot. As in the first proposed scheme, recovery mechanisms are necessary if any ballot check fails. The voter performs the following steps to vote:

1. **Selecting the option**

   In the voting booth, the voter marks her choice on the top page and it is transferred to the bottom page;

2. **Verifying the ballot**

   She then inserts her ballot into a special envelope, which has transparent borders and a window to show just the scratch surface. After this, she hands the envelope to the authorities. They verify the surface on the top page is intact and the voter did not separate the two pages;

3. **Casting the top page**

   The voter separates the pages of the ballot and casts the top page into the conventional ballot box;

4. **Obtaining the receipt**

   She casts the bottom page into the Farnel box. The box shuffles its contents and outputs copies of random selected bottom pages as receipt.

The scheme can be adapted to prevent a malicious voter from casting a ballot different from the received one. For example, the ballot can include a number that is removed before the voter casts her vote. This technique would be similar to the solution of Harris for chain voting attacks presented in Section 4.2.2.

Observe that the special envelope prevents the authorities from learning the voter's choice while verifying the surfaces and the pages were not separated before.

Tallying the Votes

As the scheme presented in the last section, the contents of the two ballot boxes are published on a bulletin board. However, the scratch surface on the top pages should be removed before publishing the pages and the commitments should be decommited to verify the ballots. That is, the random number and the options on the top page are hashed together and the resulting hash is compared with the hash on the ballot. The results are computed from the top pages.

Verifying the Votes

From the information on the board, everyone can perform the same procedures as the talliers to verify the votes. The voters, especially, match their receipts with the corresponding bottom pages on the board.

4.4.4 Evaluation

As the proposal presented before, the security of the scheme here depends on the ballot boxes and on the ballot form. Due to the new ballot design, however, the scheme does not requires trustworthy talliers.

The votes are tabulated from the top pages and the receipts are made from the bottom pages (without the scratch surfaces). Because each bottom page contains the *same* selections of its corresponding top page and also includes the commitment to the options on the top page, an adversary cannot replace a top page by another with a different permutation or with a selection for a different option, without being detected. Moreover, since the bottom page does not include the option selected, an adversary cannot use receipts to obtain indication of the results before the voting closes. Thus, the new ballot design satisfies the requirements presented in Section 4.4.1.

The ballot design contains commitments that could be saved by a malicious voter and used later to violate her privacy. These commitments, though, are covered by scratch surfaces and the authorities verify the surfaces are intact before allowing the voter to cast her vote. This way, the scheme prevents the voter from obtaining any information (except the option she selected and the ballot's permutation) from her ballot form.

A voting accomplished through this scheme can be verified by voters and third parties. Everyone can verify on the bulletin board each top page has a corresponding bottom page with the same hash and check both pages are marked in the same position. Also, by hashing the options and the key, everyone can verify the vote. Voters, especially, can match their receipts with the bottom pages on the board.

The voter privacy is preserved as well. The Farnel box anonymizes the bottom page cast by the voter and output random chosen bottom pages as receipts. This way, the voter cannot use the receipts to indicate her vote even if she receives her bottom page.

4.4.5 Extensions

The special envelope required in step 3 of the scheme above helps the authorities to verify the ballot pages were not separated before. This envelope, though, may not be sufficient to check the separation. As consequence, a malicious voter could attempt to discredit a voting by marking different options on the two pages.

In order to counter this threat, a physical mechanism to prevent the voter from separating the pages could be used, such as the folder with key used in Punchscan [77]. By means of this folder, the authorities lock the ballot inside it and just unlock it after the voter has marked her vote. Another solution requires the modification of the ballot form. Instead of using a carbon mechanism to transfer the marks, the top page would have holes à la Punchscan and the voter would use a bingo dauber to select her option. Each role, however, would have a different pattern. Thus, if the voter marks a pattern on the top page, she could not mark a different pattern on the bottom page as the patterns would not match on both pages.

4.5 Single Box Farnel Scheme

The design presented above is awkward in several aspects: it requires two ballots boxes and the vote casting procedure is rather complicated and vulnerable to certain threats. The voter should cast the two pages of her ballot in different ballot boxes. Also, the procedure of verifying the marks on the two pages could expose the voter's option if not performed correctly.

We present here an improved version of the scheme presented in the last section. The scheme requires just a Farnel box and uses a simpler vote casting procedure.

4.5.1 Requisites

The Ballot Form

The ballot form here follows the structure of the form presented in Section 4.3.1, that is, it has two pages that are initially overlaid and marks performed on one page are transfered to the other page. The top page, though, contains *only* the options in a random order along with bubbles to select them. The bottom page contains the same bubbles as the top page and an index. Also, it includes one commitment to the options of the top page and the index. The index indicates the options' order and helps the authorities to identify the order in the tallying process. The commitment and the index are printed at the foot of the page, on the left and on the middle, respectively. In addition, the bottom page includes the corresponding decommitment that is printed close to the index. The commitment is covered by a scratch surface apart from the index and the decommitment.

The new ballot design can be also described as follows. Let C be a set of options available, I a set of positive integers, π_C the permutation of C, H a secure hash function used as commitment, i an index that is a unique number in I, and r a random number from a large (key) space. The top page is composed of π_C and of the bubbles to select the options. The bottom page contains $H(\pi_C, r, i)$, r, i, and the same bubbles as the top page. Figure 4.4 illustrates the ballot form required by the scheme.

The list of possible permutations for all ballots as well as the index corresponding to each permutation are published on the bulletin board before the voting.

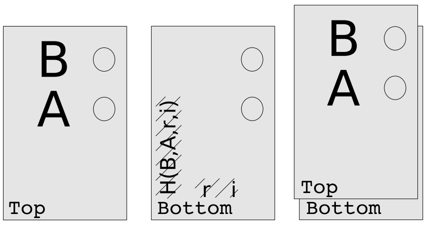


Figure 4.4: A ballot form in the single box Farnel scheme.

The Ballot Box

Differently from the previous proposals, the scheme employs just a Farnel box as described in Chapter 3.

4.5.2 The Scheme

Before the Voting

The authorities define the parameters of the Farnel box (i.e. the number of receipts per voter l and the number of dummy votes *Init*) and initialize the box with dummy votes (see Sections 3.5 and 3.4 for details).

For the initialization as well as for the voting phase, an auditing process is necessary. The audit has the goal of detecting malformed ballot forms and is performed as follows. The authorities select a set of blank ballots at random, separate the two pages of each ballot, and detach their scratch surfaces. In order to verify a ballot, the authorities hash the options on the top page (π_C) along with the random number (r) and with the index (i) printed on the bottom page. Then they compare the resulting hash with the value ($H(\pi_C, r, i)$) also on the bottom page. In addition, they verify the randomization on the top page and the randomization indicated by the index i match.

In the voting phase, helper organizations assist the voter to audit blank ballot forms in the same way. That is, the voter selects some ballots at random and hands them to the organizations that verify the commitments on the ballots.

After auditing the ballots, the authorities publicly initialize the Farnel box as follows. First, they mark an option on each (entire) blank ballot. After this, the authorities separate the two pages of the ballots and destroy the top pages through a paper shredder. Now they cast all bottom pages into the Farnel box and publish the number of votes cast per option on the bulletin board.

Voting

The voting authorities hand a blank ballot to the voter in a sealed envelope after verifying her eligibility. The voter can either use the blank ballot to vote or audit it as described above.

Assuming that the ballot form verified is well-formed, it is discarded and the authorities hand a new blank ballot to the voter. In principle, we could allow the voter to opt to audit a number of ballots before accepting one to use to cast her vote. If any ballot audit fails, recovery mechanisms have to be used. The voter performs the following steps to vote (see also Figure 4.5):

1. **Selecting the option**

   In the voting booth, the voter chooses her option on the ballot form and marks the corresponding bubble (a);

2. **Verifying the ballot**

   She separates the two pages of her ballot (b) and adds the bottom page into an envelope to make visible *only* the scratch surfaces. After this, she destroys in public the top page by using a paper shredder (c) and hands the envelope containing the bottom page to the officials. They verify the surfaces are entire;

3. **Casting the vote**

   The voter removes the bottom page from the envelope and casts it publicly into the Farnel box (d);

4. **Obtaining the receipt**

   After receiving the bottom page, the Farnel box removes the scratch surface that covers *only* the commitment value on the left side, shuffles its set of bottom pages (e), and copies l of them. The copies are held by the voter as her receipt (f).

The envelope used in step 2 allows the officials to verify the scratch surfaces without observing the marks. In principle, this envelope is not necessary and the officials could verify the bottom page directly. This verification would not expose the voter's choice as the bottom page contains the marks without their corresponding candidates. However, because a malicious official could also identify the marks, an adversary could collude with him to mount a randomization attack (see Section 4.2.2 for details). That is, the adversary could instruct the voter to mark her ballot in a determined position and ask the malicious official later if the voter has followed his instructions.

Note that the scheme may employ a mechanism to prevent voters from destroying top pages others than theirs. For example, the ballots could be numbered in a similar way as in the case of preventing chain voting attacks (see Section 4.2.2).

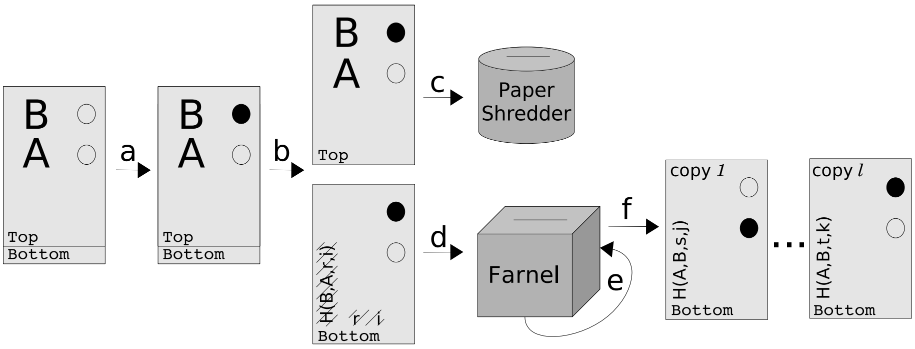


Figure 4.5: The voting steps of the single box Farnel scheme.

Recovering and Tallying the Votes

In order to count the votes, the talliers open the Farnel box, detach the scratch surfaces on all ballots, and publish the votes on the bulletin board. The talliers, then, start the process to recover the votes. In this process, they compare the index on the votes with the index on the bulletin board to identify the permutation of the options; remember that the permutations as well as their indexes were previously published. From the permutation identified and the mark on the vote, the authorities determine the option chosen by the voter.

After recovering the votes, the talliers open all commitments using the random numbers and the indexes. In this step, they hash the random number and the index along with the permutation identified before, and then compare the resulting hash with the hash on the vote. Now, the authorities count the votes in the same way as the original Farnel does, that is, all votes are counted and the votes cast during the initialization phase are subtracted from this sum. This last step is unnecessary if all initializing votes are void votes.

Verifying the Votes

Voters can, as usual, visit the bulletin board and confirm that their receipts appear accurately, and complain otherwise. In particular, they verify the commitments and the marks on their receipts correspond to those on the votes published on the board. Helper organizations and observers verify the talliers performed their work correctly.

4.5.3 Evaluation

As the solution presented in Section 4.4, the scheme relies on the ballot form, on the Farnel box, and on the bulletin board to satisfy the privacy and the verifiability properties.

The receipt fulfills the requisites described in Section 4.4.1. Each receipt is a copy of a bottom page and contains *only* the voter's choice and the commitment to the options. Because the receipts do not include the decommitment values, an adversary cannot extract the candidate selected from the receipts and thus obtain information about the results before the tallying.

The commitments and the selections on the receipts also prevent an adversary from replacing votes without being detected. Because the receipt commits the option as well as the index that points out the permutation, the adversary cannot make a fake vote using an option, a different permutation of options, or a different index. In addition, because the receipt contains a mark in the same position of the corresponding vote, the adversary cannot make a fake vote with a different mark.

The scheme is public and voter-verifiable. From the bottom page published on the board, everyone can hash the index, the options that the index points out along with the key and verify the resulting hash match the value of the page. The voters, in addition, can match the marks on their receipts with the corresponding votes on the board and verify the hash on their receipts and on the votes are the same.

The scheme meets privacy. When casting her vote into the Farnel box, the only information that the voter learns is the option that she chooses and the permutation of her ballot; the commitments as well as the decommitment values are hidden under scratch surfaces. Due to this and the fact that the Farnel box returns the voter copies of random selected votes, the voter cannot compromise her privacy.

4.5.4 Extension - Human Readable Paper Audit Trail (HRPAT)

In the manner of Ryan [84], the scheme could be adapted to provide a HRPAT by employing a conventional ballot box as alternative to the paper shredder. This way, instead of destroying the top page in a paper shredder, this page is cast into the conventional ballot box. The box would store the top pages as an audit trail so that the votes can be counted without depending on the votes from the Farnel box.

4.6 Improving the Threeballot Voting System

The ThreeBallot voting system was proposed by Rivest [80, 82]. The goal of this system is to satisfy voter-verifiability without relying on cryptography. Several drawbacks, though, have been reported for Threeballot and improvements were incorporated in its newer versions. In this section we introduce a variant of Threeballot that aims mainly at solving the information leakage problem. This problem was pointed in [5] and independently by Clark et al. [53].

4.6.1 An Overview of the ThreeBallot Voting System

The scheme employs a ballot design that consists of three single ballots. The ballots are identical except for the random *ID*s printed on the bottom of them. That is, they have the same list of options and bubbles to select them, but each ballot has a unique random *ID*. The *ID*s are encoded in a way that the voter cannot remember them. Figure 4.6 shows an example of the three single ballots.

In Threeballot, the voter should follow some rules to mark her ballot. That is, in order to vote for an option, the voter should mark the same option in two of the three ballots. The other options, though, should receive one mark each one in one of the three ballots. The choice of which of the three ballots she places these marks should otherwise be random.

After marking her three ballots, the voter inserts them into a machine that verifies the voter has marked the three ballots according to Threeballot rules. If the ballots were marked correctly, the voter chooses one of the three ballots as her receipt and the machine copies the ballot selected. Ideally, this should be done in a way that prevents the system from learning which of the three ballots the voter chose to retain as her receipt. Now, the machine casts the three original ballots into a conventional ballot box to finish the process.

At end of the voting, the ballots cast are published on a bulletin board and all votes are counted. As each option received one extra vote, these votes are subtracted from the count to obtain the final results.

Drawbacks

The version of Threeballot presented in [80] has several drawbacks as also discussed by Rivest. However, most of them were mitigated or even solved in the last

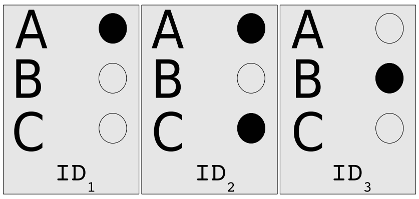


Figure 4.6: An example of a vote for option A in Threeballot.

version presented in [82]. We briefly describe here two known problems of Threeballot: the reconstruction attack and the information leakage problem. Although Rivest proposed mitigations to the former problem, the latter is still unsolved so far.

The reconstruction attack As described above, all three single ballots are published on the bulletin board after the voting. Strauss [92] showed through simulations that from the ballots published, it might be possible to reconstruct the triples and so violate the voters privacy. In order to accomplish the reconstruction, the adversary chooses a targeted single ballot (possibly the receipt of a voter) and matches it to every possible pair of ballots on the board. As result, the attacker might find either the two other ballots that compose uniquely the valid triple or a set of possible pairs that form valid triples with the targeted ballot.

In order to mitigate this attack, Rivest proposed to replace receipts by means of the (original) Farnel idea in [80], that is, instead of keeping her own receipt, the voter casts it into a Farnel like box and receives another one. In [82], Rivest et al. consider a short ballot form (i.e. a ballot with few races and few candidates per race) to increase the possibility ballots being cast with the same pattern.

The information leakage problem The voter's receipt in Threeballot is a copy of a single ballot and so it contains part of the marks of the three ballots. Although the receipt cannot be used to associate the voter to her vote, it reveals a tiny bit of information about the voter's choice. This information cannot be used against the voter. However, in a large set of receipts, statistical indication of the voting results can be exposed before the voting ends.

The flaw is best explained through an extreme example. Suppose a voting with three candidates where one of them receives all votes and the other two none. In addition, suppose that all voters behave uniformly at random with regard the marks and the column they choose as receipt. Finally, assume that all voters show their receipts to a helper organization.

Counting the number of marks for each candidate (row) on the receipts reveals information on who is winning the voting at that particular polling place. In this example, the winning candidate can expect 2/3 mark per receipt, whereas all the others can expect only 1/3 mark per receipt. The information is of a statistical nature.

To show the effect, we wrote a small simulation program. Table 4.1 shows ten simulations for an election with three candidates (1, 2, and 3), where 100 receipts have been collected and candidate 1 gets all the votes. The lines show the number of marks for each candidate, leaving no doubt at all about who is winning already while voting is still going on.

1	2	3	4	5	6	7	8	9	10
69	73	61	65	65	64	65	65	68	61
34	39	32	37	29	32	30	31	29	34
43	34	31	37	30	37	37	28	26	27

Table 4.1: A simulation of ten elections where every voter votes for candidate 1 and 100 receipts are collected per election.

In fact we are dealing with two (p, n)-Bernoulli distributions: one with $p = 2/3$, and the others with $p = 1/3$. In both cases n=#receipts.

Observe that adding candidates (rows) to the ballot does not help. Adding columns does, because it flattens the distributions ($p = 1/4$ vs. $p = 2/4$; $p = 1/5$ vs. $p = 2/5$ etc.), but this is undesirable for practical reasons. In addition, note that a statistical analysis is more difficult if the voters do not behave randomly and the original scheme is used: the voter chooses which column to copy.

This flaw in the ThreeBallot system is debatable. Although the information obtained from the receipts has the same effect as exit polls, there are two differences with regard these voting. First, not every country has or allows exit polls. Second, voters can lie about how they voted, whereas in the threeballot system the receipts reveal actual information. In an election where the difference of votes among two candidates is small, for example, the information obtained from the receipts can certainly influence voters while the election is going on.

In the next section, we present a version of Threeballot that mitigates the reconstruction attack and the information leakage problem.

4.6.2 The New Proposal

As described before, the votes in the Threeballot system can be reconstructed if no countermeasure is used and its receipt leaks statistical information about the voting results. We now introduce a version of Threeballot that overcomes these problems. Our solution can be seen as a combination of Threeballot scheme with Prêt-à-Voter [83, 22] based ballot forms and with Farnel. It employs cryptography and does not rely on a machine to check the validity of the voter selections.

Requisites

An initial ballot design As the proposal of Rivest, the ballot form is composed of three single ballots. Here, though, the options on the three ballots are permuted and every ballot has a top and a bottom pages that are initially overlaid. More specifically, the top page of each single ballot is composed of the *same* permuted list of options, bubbles to select the options, and a distinct commitment and its decommitment to the options; the corresponding bottom page has a copy of the bubbles and of the commitment printed on top page.

The initial design is also presented as follows. Let C be a list of options, π_C the permutation of C, H a secure hash function used as commitment, and a random number r_j $(1 \leq j \leq 3)$. Each single top page is composed of π_C, r_j, $H(\pi_C, r_j)$, and the bubbles. The bottom page contains only $H(\pi_C, r_j)$ and the bubbles. As before, the commitments and decommitments are covered by scratch surfaces and a carbon mechanism copies the mark from the top page to the bottom page. Figure 4.7 shows the new ballot design.

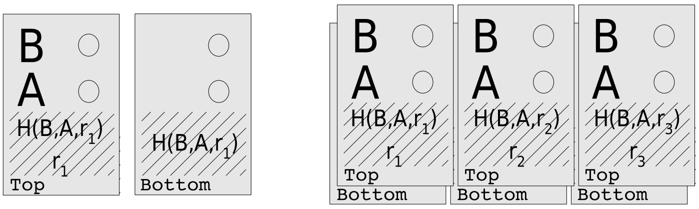


Figure 4.7: On the left the two pages of the first single ballot. On the right the three single ballots that compose the new ballot design for the Threeballot scheme.

The ballot boxes In addition to the special ballot design, the scheme employs two ballot boxes, that is, a Farnel box as described in Chapter 3 and a conventional box.

The Scheme

Before the Voting Following the Farnel idea, our proposal has a setup phase where the ballot boxes are initialized. Considering the ballot design described above and the initialization process presented in Section 3.4, the authorities initialize the conventional box with marked top pages and the Farnel box with the related marked bottom pages.

Note that, in Threeballot, it is easy to encode void votes (see Section 3.4 for more information) without needing to introduce an explicit void option: the ballot form is simply marked with exactly one bubble against each candidate. We still need to keep a protected record of how many initial, void votes are cast as this information is needed for the final check-sums.

As described in Section 3.4, the initialization process includes an audit to detect malformation of ballots (e.g. ballots with invalid commitments). In order to audit a ballot here, the scratch surfaces on both pages of the three single ballots are detached. Then, the commitments on both pages of each ballot are compared and opened using the decommitment value from the top page. Specially, the options (π_C) printed on the form are hashed along with the random number (r_j) and the result is compared with the commitment ($H(\pi_C, r_j)$) on the top page.

The auditing process is also performed by the voter in the voting phase. That is, before receiving her blank ballot, the voter assisted by helper organizations audit some blank ballots as above. Note that, in this case, a mechanism of recovery should be used if any ballot audit fails.

Voting Upon proving her eligibity to the voting authorities, the voter receives a sealed envelope with a blank ballot and performs the following steps to cast her vote:

1. **Marking the ballot**

   As the original Threeballot scheme, the voter marks her option in two of the three ballots and each other option once in one of the three ballots. The marks are transferred from the top page to the bottom page;

2. **Verifying the scratch surfaces**

   The voter inserts the three ballots, with their pages still overlaid, into three envelopes apart. These envelopes have windows to show just the scratch surfaces and are transparent on the borders. She then hands the envelopes to the authorities that verify the scratches are intact and the ballots were not separated before;

3. **Verifying the marks on the bottom pages**

   The voter separates the two pages of each ballot and hands the bottom pages to the authorities. They verify the pages were correctly marked according to the Threeballot rules;

4. **Casting the vote**

   The voter casts the three top pages of her ballot apart into the conventional ballot box;

5. **Obtaining a receipt**

   In order to obtain her receipt, the voter casts the three bottom pages of her vote into the Farnel box in the presence of the authorities. The Farnel box removes the scratch surfaces on the pages, shuffles its set of bottom pages, and returns one or more copies of random bottom pages to the voter.

Note that as the officials verify the marks on the ballot, the protocol does not prevent randomization attacks (see Section 4.2.2). A malicious official may observe the marks and reveal them to an adversary afterwards.

Tallying the Votes When the voting has finished, the talliers open in public the conventional ballot box, detach the scratch strips on all top pages from the box, and publish the votes on a bulletin board. Additionally, they also publish the bottom pages from the Farnel box. In order to compute the voting results, the authorities count all votes on the top pages, and subtract from them the votes cast before the voting and the extra votes cast à la Threeballot scheme.

Verifying the Votes From the top pages posted on the board, anyone verifies if the decommitment values open their respective commitments. As before, this is performed by hashing the options along with the random number, and then comparing the result with the original hash. In addition, the voters can search top pages on the board that correspond to their receipts (i.e. copies of bottom pages).

Extensions

The scheme could use the original Farnel ballot box (see Chapter 3) adapted to remove scratch surfaces in replacement of the enhanced Farnel box. In this case, the box would only exchange receipts as in [80]. That is, instead of casting the three bottom pages of her ballot (see step 5 above), the voter chooses one of the three pages, casts it into the adapted box, and destroys the other two pages; the box gives the voter a random selected receipt from its set. Alternatively, the voter could cast her three bottom pages and receive three random pages from the Farnel box. These alternatives have the advantage of employing a more simple Farnel box while not requiring a check machine.

4.6.3 Achieving Ballot Secrecy without the Farnel Box

The version of the Threeballot scheme proposed above does not leak statistical information. On the other hand, it actually makes the reconstruction attack more virulent, but this is countered by the Farnel mechanism. It, however, relies on Farnel ballot box that adds complexity to the scheme. We discuss now some ideas to simplify the scheme by making it independent of the Farnel box. Notice that the ideas are only of theoretical interest.

At first glance, one might think that the new ballot design could be enough to accomplish a scheme without the Farnel box. Indeed, the new design would overcome the information leakage problem. However, it would make the reconstruction attack easier than in the original Theeballot scheme. Because the ballot forms have different permutations and each form employs the same permutation in the three single ballots, the bulletin board can be segmented in groups according the permutations used. For a number of voters N and a number of candidates C, the number of groups is $3n/C!$ or $3n/C$ for cyclic permutations. This way, an attacker only needs to compare ballots in the same group to reconstruct the votes.

In order to render the reconstruction attack harder to perform, we change our ballot design to allow distinct permutations in the same ballot. That is, instead of using the same permutation in the three ballots, a random permutation is selected for each of the three single ballots. However, now some voters can have problems to select options in different positions in the three ballots. In addition, the authorities cannot verify the marks on the three bottom pages as the options can be in different positions. The first problem could be reduced by training the voters[1], so we concentrate in a solution for the second one.

A possible solution for verifying the marks on a ballot is to introduce a mapping between the two pages. That is, the options on the top page are mapped to elements on the bottom page and each option can be associated to different elements in other ballots (see Figure 4.8 for an example). Thus, the authorities can verify the bottom page without recognizing the options. This mapping could be, for example, a bijection between the set of options and a set of positive integers.

Although the mapping hides the options from the authorities and still makes possible the check, it would reveal the relationship between the three permutations to the authorities. This way, a malicious authority could perform the reconstruction attack by grouping the permutations as before.

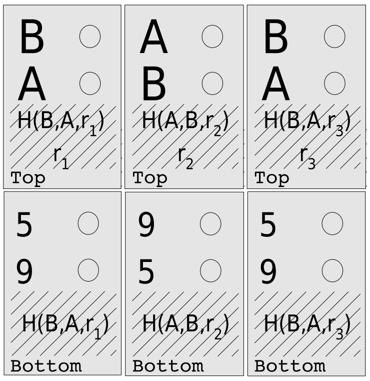


Figure 4.8: An example of ballot with mapping.

[1] Note that despite the training, the scheme is not simple and the voters would still have trouble to fill out their ballots.

The mapping described makes the scheme simple as the authorities can verify the marks, but it is insecure. Thus, we rely on a check machine to solve the problem such as the original Threeballot scheme and avoid the mapping. The drawback is that here the machine needs to identify the options to verify the marks.

We assume, however, that the machine can perform comparisons without storing the options and describe the ballot form as follows: let C be a set of options, π_C^j $(1 \leq j \leq 3)$ be the permutation of C, H be a secure hash function, and a random number r_j $(1 \leq j \leq 3)$ from a large (key) space. Each single top page is composed of π_C^j, r_j, $H(\pi_C, r_j)$ along with the bubbles to mark the options. The corresponding bottom page contains the same $H(\pi_C^j, r_j)$ and bubbles of the top pages. Figure 4.9 illustrates this ballot design.

Based on this modified ballot form and considering the check machine to verify the votes, the voter now performs the following steps to vote:

1. **Selecting the option**

   She marks her vote on her ballot and verifies the marks by using the check machine as the Threeballot scheme;

2. **Verifying the ballot**

   The voter inserts her ballot into an envelope that hides the options and hands it to the authorities. The authorities verify the ballot was not separated and check the scratches on the top pages;

3. **Casting the vote**

   She separates her ballot and casts the three single top pages apart into the conventional ballot box;

4. **Obtaining the receipt**

   In the presence of the authorities, the voter keeps one of the three bottom pages as receipt and destroys the other two in a paper shredder.

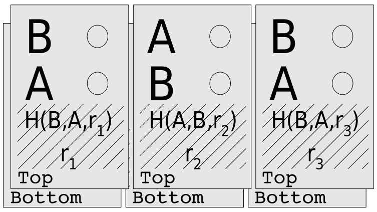


Figure 4.9: A ballot design for three ballots with different permutations.

As we changed just the structure of the ballot by employing different permutations, the tallying of votes remains the same described in Section 4.6.2.

4.7 Randell-Ryan Scheme with Farnel

Randell and Ryan [78] proposed a simple scheme that intends to improve the voter secrecy in the existing manual systems and to provide voter-verifiability. The scheme is based on Prêt-à-Voter [83, 22] ballot forms and does not rely on cryptography. It, however, has a drawback similar to the scheme presented in Section 4.3. That is, it requires honest authorities to prevent manipulation during the tally.

In this section we introduce a version of the Randell-Ryan scheme that employs a Farnel box and a modified ballot form. The new scheme removes the drawback of the original version.

4.7.1 An Overview of Randell-Ryan Voting Scheme

The scheme employs a ballot form that has a left (LC) and a right (RC) column. These columns are separated vertically by a perforation. The LC contains a randomized options list and the RC has the respective spaces to select the options. A serial number, CIN (code identification number), identifies the ballot uniquely and is printed at the foot of the LC. The RC has the same CIN at its foot, but it is covered by a scratch surface; this scratch has a receipt identification number (RIN) printed over it. The proposal also requires two conventional ballot boxes. One for receiving left columns and other for receiving right columns.

In order to vote, the voter selects an option on the right column. She then drops the LC into the ballot box for left columns. After that, in the presence of officials and observers, a photocopy of the right column, with the scratch strip intact, is made as the receipt and the voter drops the original RC into the other ballot box.

In the tabulation process, the officials open the ballot boxes and publish the RCs on the bulletin board so that the voters can verify their votes. The officials then scratch off the strip on the RCs to reveal the CIN. After that, they match the CIN number on the LC and on the RC columns to identify the votes.

Drawback

The scheme has a drawback similar to that of the single scheme presented in Section 4.3. That is, it requires trustworthy talliers. Otherwise, votes could be altered before being counted and the voters could not detect this.

According to the scheme, the voter receipt is a copy of the right column with its RIN. The voter compares the RIN and the mark on her receipt with the right column published on the bulletin board. However, after the authorities scratch off the RIN strips to show the CIN, she cannot verify her vote. This way, after exposing the CIN to count the votes, a malicious authority could potentially alter votes without being detected.

Of course, various mechanisms can be proposed to counter this threat, but the fact remains that the resulting levels of assurance will inevitably be lower than those achievable using cryptographic mixing/tabulating mechanisms. Thus, stronger trust assumptions in the Randell-Ryan than in Prêt-à-Voter are necessary.

4.7.2 An Improved Scheme

We present now an improved scheme that overcomes the drawback of the Randell-Ryan proposal.

Requisites

The ballot design The ballot form here is similar to that one employed in the original scheme. That is, it is composed of a left and a right column that are separated vertically by perforations. However, now the left column contains a commitment to the options and its decommitment. The right column contains spaces to select the options as well as a copy of the commitment. Scratch surfaces cover the commitments on both sides and the decommitment values.

This ballot design is additionally presented as follows. Let C be a set of options, π_C the permutation of C, H a secure hash function, and r a random number from a large (key) space. π_C, $H(\pi_C, r)$ and r compose the left column. $H(\pi_C, r)$ and spaces to select the options compose the right column. Figure 4.10 illustrates this ballot form.

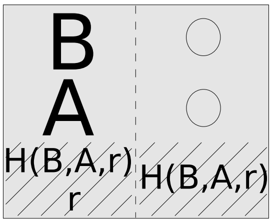


Figure 4.10: The new ballot form based on Randell-Ryan proposal.

Ballot boxes As in the Randell-Ryan scheme, we employ two ballot boxes. However, one of them is a Farnel ballot box (see Chapter 3 for details). The other is a conventional box.

The Scheme

Before the Voting Due to the Farnel ballot box, the scheme has a pre-voting phase where the Farnel box as well as the conventional box are initialized. This process is similar to that described in Section 3.4. Here, though, the conventional ballot box is initialized with left columns and the Farnel box receives the corresponding marked right columns.

In order to audit blank ballots, the scratch surfaces on both sides of the ballot are detached. By doing this, the commitments ($H(\pi_C, r)$) on both sides can be compared. After this comparison, the commitment on the left column is decommited using the random number printed on the bottom of this page. This is performed by hashing the options on the left column along with the random number. The resulting hash is compared with the values on the pages.

Voting After being authenticated by the officials, the voter receives her blank ballot in a sealed envelope. She can either audit her ballot or use it to vote. In the former case, she performs the auditing steps described above with the support of helper organizations and receives a new blank ballot from the officials after this. If any ballot audit fails, recovery mechanisms are necessary. The voter performs the following steps to vote:

1. **Marking the ballot**

   In the voting booth, the voter looks her option on the left column and marks the corresponding space on the right column as in Randell-Ryan scheme;

2. **Checking the ballot**

   She separates the left and right columns of her ballot form and inserts them into two envelopes apart. The envelopes have a window on the foot that shows only the scratch surfaces. The voter hands these envelopes to the officials that verify the surfaces are intact;

3. **Casting the vote**

   In the presence of the officials and observers, the voter casts the envelope containing the left column into the conventional box;

4. **Obtaining the receipt**

   She now removes the right column from the envelope and casts it into the Farnel box. The box then removes the scratch surface on the column and returns her copies of random chosen columns as receipt.

Tallying the Votes After the voting, the talliers open in public the two ballot boxes, and detach the scratch surfaces on all left columns (LC). Then, they open all commitments on the left columns to verify the votes. As before, this is performed by hashing the options and the random number, and then comparing the result with the hash on the pages. After that, the authorities use the hash values to match the right and left sides. This allows the marks on the RC columns to be interpreted and counted. Then, the results are computed as in the Farnel scheme. That is, all votes are counted and initial votes are subtracted from the count.

Notice that the scheme can be implemented using a ballot style without the hashes. However, in this case it may be possible for the tabulating authorities to manipulate the candidate lists unless appropriate anti-counterfeiting measures are in place. A nice feature of dropping the hashes is that we do not have to worry about checking the correctness of the hashes: a ballot is well formed as long as the CIN #s match. Thus, if we have on-demand printing of the candidates lists we simply need to ensure that the ordering is randomized.

If the threat of manipulation of the candidate's order during tabulation is regarded as sufficiently serious, then the hash function would be used. Here, though, we do need to check that the ballots are well-formed w.r.t. the hashes.

Verifying the Votes Anyone can visit the bulletin board and verify each left column has a corresponding right column with the same hash value. Moreover, anyone can make the hashes again from the options on the left columns and the key values to verify the hashes are correct. The voters match their receipts to the right columns on the board to verify the corresponding votes.

4.8 Conclusions

In this chapter we have first presented a new voter-verifiable voting scheme. This solution does not employ cryptography, but it requires trustworthy talliers to achieve security. We then have introduced a scheme that requires a commitment scheme and uses a Prêt-à-Voter style ballot form. This scheme provides better security guarantees as the first one. Our third solution simplifies the second one. It employs just a Farnel box instead of two boxes as the previous solutions.

In addition to these solutions, we have proposed improvements to the Threeballot scheme. In particular, we have combined the Threeballot along with Prêt-à-Voter style forms and the Farnel box. A version of Threeballot that does not require the Farnel box was given, but it requires a special verification machine. We have also presented a version of Randell-Ryan scheme that uses the Farnel box and provides better security assurances.

Our solutions consider that corruption of votes take place only after the voting. Because the voting process is usually supervised by third parties and part of the authorities are supposed to be honest, malicious behaviors can be identified and prevented. If we consider that the voting process is not supervised and that all authorities are dishonest, these officials could open the Farnel box and corrupt votes. The corruption may not be detected if performed during the voting and most of the votes do not have corresponding receipts. However, if all officials are dishonest, they may also collude with vote-buyers and let them observe their victims while they are voting. Moreover, they may even allow chain voting by ignoring the verification process (see Section 4.2.2). This would undermine the security of other voting systems as well. Therefore, at least a small number of honest authorities is necessary during the voting.

Chapter 5

A Coercion-Resistant Scheme for Internet Elections

This chapter contains work published in Frontiers of Electronic Voting [6] and will appear on Towards Trustworthy Election Systems book [21]. This is joint work with Sébastien Foulle and Jacques Traoré.

5.1 Introduction

Coercion and vote-selling are known threats inherent in remote voting systems. Until recently, these problems were not treated accordingly in the literature. Proposals as [8, 48, 65] prevent voters from proving by means of receipts how they have voted, but cannot avoid other threats as voters handing their private data (e.g. secret keys) to coercers or vote-buyers. Since the introduction of the concept of coercion-resistance, remote voting became more realistic. This concept was proposed by Juels, Catalano, and Jakobsson [60] (JCJ) and is the strongest known property for such scenarios. It considers not only that the voter may be coerced to prove her vote, but also that she could be forced to abstain from voting, to reveal her private data, or to cast random votes.

Along with the coercion-resistance concept, JCJ also introduced the first scheme to satisfy it. The scheme employs anonymous credentials and basically mitigates coercive attacks by allowing the voter to deceive adversaries about her true vote intention. Despite being the best solution for remote voting when compared with previous protocols, the scheme is impractical for voting with a large number of voters as it requires a quadratic work factor (in number of votes) to compute the voting results. This drawback is directly related to the fact that the scheme employs an inefficient blind comparison mechanism.

With the goal of overcoming the drawback of JCJ solution, we introduce in this chapter a novel coercion-resistant voting scheme based on JCJ ideas. The new proposal employs anonymous credentials and fight coercive attacks as the previous solution. It, however, employs special credentials and does not rely on blind comparisons. The new protocol has linear work factor and hence is efficient for large scale voting.

This chapter is organized as follows: the next section introduces the coercion-resistant property as well as the security model that the new scheme uses; Section 5.3 recalls briefly the proposal of JCJ; Section 5.4 presents the new coercion-resistant scheme; after that, an analysis of the scheme is showed in Section 5.5; finally, the chapter is concluded in Section 5.6.

5.1.1 Related Work

The proposal of Juels, Catalano, and Jakobsson first appeared in 2002 at Cryptology ePrint Archive [59]. After improvements, it was effectively published in 2005 at WPES [60].

Following JCJ's work, a number of other schemes were proposed. Acquisti [1] introduced a scheme in which the vote and the credential are combined, and which relies strongly on homomorphisms. Clarkson et al. [25] presented a variant of Prêt-à-Voter scheme suitable for Internet voting and based on decryption mixnets. Schweisgut [89] and more recently Clarkson et al. [23, 24] proposed schemes which mitigate the inefficiency problem of the JCJ solution. The former scheme relies on decryption mixnets and on tamper-resistant hardware, whereas the latter is a modified version of the JCJ proposal.

The most promissing scheme based on JCJ ideas was introduced by Smith [91]. He presented an efficient scheme with linear work factor. Weber et al. [95], however, pointed out problems of Smith's proposal and presented a protocol that combines the ideas of JCJ with a variant of Smith's mechanism. The solutions of Smith and Weber et al. are not coercion-resistant as showed in [6]. This was also noted independently by Clarkson et al. [23, 24].

The scheme that we introduced in this chapter is related to all these works since it is also based on JCJ's ideas and aims at satisfying the coercion-resistant property. We do not consider here other proposals that fulfill less strong properties as receipt-freeness.

5.2 Preliminaries

We recall next the coercion-resistant property of JCJ and present the attack model that our scheme requires. This model was used by JCJ in their coercion-resistant scheme.

5.2.1 The Coercion-Resistant Security Property

Until recently voting schemes had the goal of satisfying the receipt-freeness security property, such as the proposals of Baudron et al. [8] and Hirt-Sako [48]. That is, the notion of preventing the voter to make or obtain information to prove how she has voted.

Juels, Catalano, and Jakobsson [60], however, observed that the receipt-freeness property is not sufficient to prevent real-world attacks and introduced the concept of coercion-resistance. This concept considers not only the receipts-freeness but also the following attacks: randomization, forced-abstention, and simulation.

In the randomization attack [88], as described in Section 4.2.2, the adversary forces the voter to cast (publish) a randomly formed vote. The voter or the adversary may not know the candidate selected; the objective is to nullify the voter's intention with high probability.

An adversary may also force the voter to abstain from voting, that is the forced-abstention attack. In this case, the adversary uses any public information posted by the voter and necessary for her authentication to mount the attack. For example, if the voter is required to post a digital signature along with her vote on the bulletin board, the attacker can force the voter to not to vote and verify she followed his instructions.

In the simulation attack, an adversary forces the voter to reveal any private information that she may use to vote and vote on her behalf. For example, if the voter has a pair of keys and need to use her private key to vote, the coercer can force the voter to hand him this key.

A formal definition of the coercion-resistance property can be found in Juels, Catalano, and Jakobsson [60] and in Delaune et al. [32].

5.2.2 Attack Model

The coercion-resistance property takes into account typical problems of remote voting. The scheme we introduce below aims at satisfying this property. In order to fulfill it, however, our proposal depends on certain assumptions and conditions described here. Similarly to JCJ scheme, we consider the following:

- An adversary may impede the voter to vote, force the voter to post a random composed ballot material, or demand secret information from the voter. The adversary has limited computational power and is able to compromise only a small number of authorities;

- An adversary may monitor the voter or interact with her during the voting process. However, we suppose the adversary is not able to monitor or interact with the voter continuously during the voting period;

- A registration phase free of adversaries. That is, the registration official is trustworthy and voters receive private data securely. Also, we assume that the voters communicate with the registrar via a untappable channel and without the interference of adversaries. This channel provides information-theoretical secrecy to the communication;

- Some anonymous channels in the voting phase. These channels are used by the voters to post their votes and prevent adversaries from learning who sent a specific vote. In practice, voters may use public computers to achieve this;

- Voters cast their votes by means of reliable machines. We do not consider attacks where the adversary may control the voters' computers (e.g. by means of malwares) in order to obtain their votes or other private data. Solutions as the code voting method proposed by Joaquim et al. [54] could help to prevent these threats;

- Denial of service attacks are not considered. The scheme employs bulletin boards that receive data from anyone and hence would be susceptible to these attacks. However, challenge-response techniques like CAPTCHAS [94] or cryptographic puzzles [58] could be used to mitigate these problems.

5.3 An Overview of JCJ Scheme

The scheme of Juels, Catalano, and Jakobsson [60] relies essentially on a method of indirect identification through anonymous credentials to overcome coercive attacks. Especially, the voter receives a valid credential (e.g. an alphanumeric string) in a secure way and uses it when she want to cast her valid vote. A voter under coercion, though, is able to make a fake credential and to hand it to a coercer. When in safety, the voter can vote using her valid credential. After the end of the voting, a blind comparison mechanism distinguishes the valid credentials from the fake ones to identify the valid votes; conversely, an adversary has no way to perform this distinction.

The scheme considers a registration phase free of adversaries and a bulletin board communication model. Also, it requires the following cryptographic tools: non-interactive zero-knowledge proofs, a probabilistic threshold public-key cryptosystem, and universally verifiable mixnets. In particular, the scheme employs a plaintext equivalence test (see Section 2.5.5 for details). Loosely speaking, this primitive takes two ciphertexts as input and returns a bit indicating if the corresponding plaintexts are equal or not. The JCJ solution is briefly described as follows:

Registration phase

In this phase a trustworthy authority issues a unique valid credential, which is a random value, for each eligible voter and publishes a probabilistic encryption of each credential on the bulletin board. Let $L1$ be the list containing all credential ciphertexts published by the authority on the bulletin board.

Voting phase

In order to vote, a voter sends the following data to the bulletin board through an anonymous channel: a tuple containing her encrypted vote, her encrypted credential, and zero-knowledge proofs that the vote is for a valid candidate and that the voter knows the vote and the credential encrypted.

Tallying phase

At the end of the voting, the talliers verify all proofs posted on the board and exclude tuples with invalid proofs. From the remaining tuples, they perform a pairwise blind comparison by means of the plaintext equivalence test to remove tuples with duplicated credentials. After removing the duplicates keeping the last posted tuples, the remaining pairs of ciphertexts (a vote and a credential) form the list $L2$ and this list is sent to a mixnet. The mixnet returns $L2'$. Then, the list $L1$ created during the registration phase is sent to a different mixnet that returns $L1'$. Now, the plaintext equivalence test is used a second time to compare (pairwise) the credentials on the list $L1'$ with the credentials on the list $L2'$. A vote is removed if its encrypted credential in $L2'$ does not match with an element of $L1'$. Finally, the votes with valid credentials are decrypted by the talliers.

Drawback

Although the JCJ scheme fulfills the coercion-resistance property, the pairwise blind comparisons involving the plaintext equivalence tests makes it inefficient for large scale elections. Let N be the number of voters and V be the number of posted votes, one has $V \geq N$ and the overhead to perform the tests is quadratic in V.

5.4 The New Scheme

The voting scheme introduced in this chapter inherits the concept presented by JCJ in their scheme. That is, voters employ anonymous credentials to cast their votes and are not directly identified in the voting process. Moreover, voters may vote multiple times and deceive adversaries by means of fake credentials. Our proposal, though, differs from the previous solution basically in two ways: it relies on special formed credentials and does not require blind comparisons against a list of encrypted credentials.

Our proposal is summarized as follows: in a registration phase, trustworthy registrars issue securely to each eligible voter a valid credential that is composed of four parts. This is performed without storing the credential in an encrypted form for later comparison as in the JCJ scheme. At time of voting, the voter performs computations and publishes ciphertexts corresponding to the parts of her credential as well as to her vote and a set of zero-knowledge proofs on the bulletin board. In the tallying phase, votes with invalid proofs and all votes posted with the same credential but the last posted are excluded. After mixing the remaining encrypted votes along with their credentials, the talliers assisted by the registrars apply a function to parts of each credential to identify the valid votes. Finally, the votes with valid credentials are decrypted.

We introduce next the details regarding the new coercion-resistant scheme.

5.4.1 Credentials

In the proposal of JCJ, a valid credential is a random string. The scheme proposed here, differently, employs credentials that has a mathematical structure based on the group signature scheme of Camenisch and Lysyanskaya [17].

A valid credential in our scheme has the following form: let $\mathbb{G}$ be a cyclic group with prime order p where the decision Diffie-Hellman (DDH) problem is hard, (x, y) two secret keys, a a random number in $\mathbb{G}$ (with $a \neq 1$), r a random number in Z_p, the credential is composed of $(r, a, b = a^y, c = a^{x+rxy})$.

The security of our credentials relies heavily on the LRSW (Lysyanskaya, Rivest, Sahai, and Wolf) and on the DDH assumptions (see Section 2.2 for details). The former assumption ensures that even if an adversary has many genuine credentials (r_i, a_i, b_i, c_i), it is hard for him to forge a new and valid credential (r, a, b, c), with $r \neq r_i$ for all i. This assumption is known to hold for generic groups and the security of Camenisch and Lysyanskaya's signature scheme also relies on it. The DDH assumption ensures that the voter cannot prove to anyone else whether (r, a, b, c) is a valid credential or not. This way, a voter under coercion can make a fake r to deceive an adversary who will not be able to distinguish between a fake and a valid r.

In a real-world scenario, our credential can be seen as containing two parts: a short one, that is r, which must be kept secret, and a long one, that is (a, b, c). The first part (i.e. r) has around twenty ASCII characters (this corresponds to 160 bits, the actual secure size for the order of generic groups), so a small piece of paper and a pen are sufficient to write r down. The other part can be stored in a device or be even sent by email to the voter without compromising the credential security.

5.4.2 Participants and Notation

The scheme involves three participants.

Voter The voter is denoted by V. She holds a valid credential and uses it when she wants to cast her valid vote. Also, she is able to make fake credentials and use them to deceive adversaries;

Talliers These authorities are represented by T. They are responsible for controlling the bulletin board, for running the mixnet, and for computing the voting results. They share an M-El Gamal private key $\widehat{T}$ corresponding to a public key T;

Registrars They are denoted by R. These authorities are responsible for issuing a valid credential for each eligible voter. Also, they help the talliers to identify valid credentials. They share two private keys, x and y corresponding to the public keys R_x and R_y.

We employ the following notation in the description below: BB is a bulletin board, $E_T[m]$ is an M-El Gamal encryption of a message m constructed with T, and $D_{\hat{T}}[m]$ is an M-El Gamal decryption of m with $\widehat{T}$. See Section 2.3.1 for a description of this version of the El Gamal cryptosystem.

5.4.3 Scheme in Details

Setup phase

In this phase the general voting parameters are established and published along with a digital signature on BB. These parameters consist of a cyclic group $\mathbb{G}$ with prime order p where the decision Diffie-Hellman problem is hard, a random generator o of $\mathbb{G}$ as well as the keys of T. Especially, the talliers T cooperate to generate the public key T and the shared private key $\widehat{T}$ via the M-El Gamal threshold cryptosystem (see Section 2.3.1).

The registrars R collaborate to produce their public keys R_x and R_y and their respective shared private keys x and y. These public keys are computed as follows: $R_x = g^x$ and $R_y = g^y$, where g is a public random generator of $\mathbb{G}$. This is accomplished through the key generation protocol described in Section 2.3.1. Also, the list of voting candidates available is published.

Registration phase

After verifying that a voter is eligible, R issues to the voter a secret credential $\sigma = (r, a, b, c)$ via an untappable channel, where a is a random element in $\mathbb{G}$ (with $a \neq 1$), r is a random element in Z_p, $b = a^y$, and $c = a^{x+rxy}$. In addition, R may furnish the voter with a designated verifier proof [51] of well-formedness for σ. By means of this proof, the voter can check the validity of his credential, but cannot convince others about this.

Note that if (r, a, b, c) is valid, then for all r the credential (r, a^l, b^l, c^l) for $l \in_R Z_p$ is a valid one too. This property is used by the voter to change the values $a, b,$ and c each time she votes.

Voting phase

The voter casts her ballot by sending the following tuple through an anonymous channel to BB:

$$\langle E_T[C], a, E_T[a^r], E_T[a^{ry}], E_T[a^{x+rxy}], o^r, P \rangle$$

C is the candidate chosen, $(r, a, a^r, a^{ry}, a^{x+rxy})$ correspond to voter's credential, o is the public generator of $\mathbb{G}$ published in the setup phase, and P is a list of non-interactive zero-knowledge proofs which ensures that the vote is well-formed. In particular, P contains a proof that the vote is for a valid candidate as presented in Section 2.5.6, proofs of knowledge of the plaintexts according to Section 2.5.1, and a proof that $E_T[a^r]$ and o^r contain the same r as described in Section 2.5.4.

Recall from the previous paragraph that the values a, $b = a^y$, and $c = a^{x+rxy}$ have been changed by the voter and are therefore different from the ones she received from R.

The value o^r is used to detect duplicates and guarantees that only one vote per voter will be counted. Otherwise, a dishonest voter could vote several times without being detected.

Tallying phase

In order to compute the voting results, the talliers T perform the following steps:

1. **Verifying proofs**

   T verifies the proofs P on each tuple and removes tuples with invalid proofs. That is, T verifies that a is in $\mathbb{G}$ and $a \neq 1$, that $E_T[C]$ is a vote for a valid candidate, the proofs of knowledge of the plaintexts, and the proof that $E_T[a^r]$ and o^r contain the same r;

2. **Removing duplicates**

   In order to exclude duplicates, T first identifies them by comparing all o^r, for instance, using a hashtable [27]. After this, T keeps the last posted tuples based on the order of posting on the bulletin board;

3. **Encrypting the plaintext element**

   The tuples that passed the previous steps have their values o^r and P deleted, and their second component (i.e. a) replaced by the M-El Gamal ciphertext $E_T[a]$. This way, only the values $E_T[C], E_T[a], E_T[a^r], E_T[a^{ry}], E_T[a^{x+rxy}]$ are processed in the next step;

4. **Mixing tuples**

   T sends the tuples composed of $E_T[C], E_T[a], E_T[a^r], E_T[a^{ry}]$, $E_T[a^{x+rxy}]$ to a universally verifiable mixnet and publish the output on BB. Let the tuples formed by $(t, u, v, w, z) = (E_T[C]', E_T[a]', E_T[a^r]', E_T[a^{ry}]', E_T[a^{x+rxy}]')$ be the mixnet output, where $E_T[X]'$ means a reencryption of $E_T[X]$;

5. **Identifying valid votes**

    a) For each tuple, R first employs its secret key y to cooperatively compute v^y;

    b) By means of the plaintext equivalence test from Section 2.5.5, R checks whether v^y and w have the same plaintext;

    c) If the verification result is positive, R generates a fresh shared key $\alpha \in_R Z_p$ and cooperatively computes $(zu^{-x}w^{-x})^\alpha$ using the shared private key x that was generated along with y in the setup phase. The new key is generated using the key generation protocol from Section 2.3.1;

    d) The officials T collaborate to decrypt the resulting ciphertext processed by R through the decryption protocol presented in Section 2.3.1. The decryption is equal to 1 if and only if the credential is a valid one. Note that if the credential is invalid, just computing and decrypting $(zu^{-x}w^{-x})$ may give some information to an adversary, so the random exponent α is necessary;

6. **Decrypting and counting the votes**

    T employs its shared private key $\widehat{T}$ to cooperatively decrypt $E_T[C]$ of each tuple with a valid credential. After that, they count the votes and publish the results on BB.

Notice that a voter under coercion should reveal the correct values a and b of her credential. Otherwise, an adversary can test whether this pair is correct by mounting a ”1009 attack” [91]. That is, the adversary sends ”1009” ballots containing pairs of the form (a^{li}, b^{li}) using 1009 random values li and checks whether more than 1009 ballots passed the first test in step 5 of the tallying phase.

5.4.4 Multiple Elections

The number of eligible voters may change in different elections. Some voters may have their right to vote revoked after having participated in an election, for instance. Also, a voter may be allowed to vote in several elections, but may not in others. In order to satisfy these scenarios, a credential is normally required to be used in multiple elections and should be revoked by the authorities when necessary.

The credential we proposed may be used in multiple elections as long as the same keys (x, y) are employed. However, in principle a credential cannot be revoked. As only the voters know their credentials, the authorities are not able to revoke a credential. In addition, even if the authorities store all credentials issued, they are not able to efficiently identify a revoked credential since the credentials are published in an encrypted form.

Although the design of our scheme makes revocation difficult, the scheme has some properties that help accomplishing this. Upon registering, a voter receives $(r, a, b = a^y, \text{and } c = a^{x+rxy})$. As stated before, the element r must be transmitted via an untappable channel. However, the elements $(a, b = a^y, \text{and } c = a^{x+rxy})$ may be sent by post or even by email; this does not compromise the credential security as long as the DDH assumption holds. Based on this, we suggest the following method to revoke credentials and to perform new elections:

Besides generating and issuing a credential for each voter, the registrars R cooperatively compute the encryption of (a^r) and (a) (i.e. $E_R[a], E_R[a^r]$) and stores them in a list. These encryptions are performed using a public key R corresponding to a shared private key $\widehat{R}$ especially generated for this purpose.

For each new election, instead of using the same keys (x, y), the registrars generate new keys (x', y') and furnish the voters with new values $(a' = a^l, b' = a'^{y'}, \text{and } c' = a'^{x'+rx'y'})$, computed from $E_R[a]$ and $E_R[a^r]$, for a randomly chosen l. That is, c' is computed by raising $E_R[a]$ and $E_R[a^r]$ to x' and to $x'y'$ respectively, and then by using homomorphism to obtain $E_R[a^{x'+rx'y'}]$. After that, $E_R[a^{x'+rx'y'}]$ is raised to l and cooperatively decrypted. The values a' and b' can be obtained similarly, but without using homomorphism. The new elements of the credential could be sent by mail to the voter or published on a dedicated website.

5.5 Evaluation

The scheme presented in the previous section aims at fulfilling the coercion-resistant requirement as well as standard voting security requirements. We evaluate now our scheme based on these requirements and consider the attack model introduced in Section 5.2.2.

Coercion Resistance

In order to be coercion resistant, a voting scheme must be receipt-free and defeat coercive attacks, such as randomization, forced-abstention, and simulation attacks, as defined by JCJ.

A scheme is receipt-free if the voter is not able to make or obtain a receipt to prove in which way she has voted. Especially, the voter here may not convince an adversary that her credential is valid and that she used it to cast a particular vote. Our proposal satisfies these requisites. The voter is not able to prove an adversary that her credential is valid and an adversary cannot determine whether a credential is valid or not unless he can break the DDH problem. In addition, the credentials are verified only after a mixing process and the method employed to verify them (see step 5 in the tallying phase) does not leak any information. This way, the voter is not able to obtain any evidence that can be used as a proof.

The proposal we presented is resistant to the randomization attack as well. In this attack an adversary forces the voter to cast a ballot composed of random information. As the voter in our scheme publishes her vote along with a set of zero-knowledge proofs and all votes with invalid proofs are excluded, ballots randomly composed will not be tallied. In addition, even if the adversary observes the voter and forces her to vote for a random candidate, she cannot verify the voter performed this using her valid credential.

In the forced-abstention attack an adversary forces the voter to abstain from voting. This attack is possible if the adversary can verify the voter has voted. Our scheme, however, does not reveal any information about the voter identity. The voter receives a valid credential that identifies her, but it is kept hidden from adversaries. That is, the voter publishes the credential ciphertext on the bulletin board via an anonymous channel and the credential is verified in the tallying phase (step 5) without being decrypted. Hence, the adversary cannot check whether the voter has voted or not.

The fact that the voter's identity is concealed also prevent an adversary from forcing a voter to show the random exponents used for encrypting her ballot components. As the voter posts her ballot through an anonymous channel and no information about the credential is revealed during the tallying, the adversary does not know who voted. This way, a coerced voter can say an adversary that she did not vote and he cannot verify whether the voter told him the truth or not. An adversary could also force the voter to reveal the exponents before she sends her ciphertexts. However, the voter can use a fake credential and show the exponents of the corresponding components.

Our scheme also prevents the simulation attack. In this attack an adversary forces the voter to reveal her valid credential and vote on his behalf. However, the voter in our solution is able to deceive the adversary by handing him a fake credential and the adversary cannot distinguish a valid credential from a fake one under the DDH assumption. The credential structure, the mix process as well as the method used to identify valid credentials prevent the adversary from performing the distinction.

Democracy and Accuracy

In our proposal, the bulletin board may accept votes from eligible and non-eligible voters and the voters may vote multiple times. However, only votes from eligible voters appear in the final tally and only one vote per eligible voter is counted. The scheme accomplishes this by excluding votes posted with the same credential (see step 3 in tallying phase). This way, even if a voter uses the same credential to vote many times, only the last vote will be processed. In addition, the scheme checks whether the credentials are valid or not and excludes votes with fake credentials. This is performed by the method that identifies valid credentials in step 5 of the tallying phase. Since the method only outputs the value 1 for valid credentials and that it is hard to forge valid credentials under the LRSW assumption, it ensures that only votes from eligible voters will be in the final tally. Conversely, the method outputs a random number as result for invalid credentials. This way, votes from non-eligible voters (i.e. invalid votes) will not be counted.

Universal Verifiability

Anyone is able to verify the correctness of the voting process and its results in our solution. This requirement is ensured by the public bulletin board, which is secure, and by the non-interactive zero-knowledge proofs (NIZKPs). The proofs generated in all phases of the scheme are published on the bulletin board to allow anyone to verify them. In addition, the voters publish their votes on bulletin board, so anyone is able to verify the votes that will be processed. In the tallying phase, the steps performed can also be verified by anyone through the bulletin board; this includes the shuffle performed by the mixnet and our method to identify valid credentials.

The bulletin board and the NIZKPs also prevent the disassociation of the pair of ciphertexts (a vote and a credential). After the voter publishes her ballot on the board, any transformation of the ciphertexts (i.e. reencryptions) is proved through the NIZKPs.

Efficiency

As described before, the JCJ scheme requires a quadratic running time. The reason for this is the pairwise blind comparison mechanism used for removing duplicates and for identifying valid credentials. Our proposal, differently, does not rely on blind comparisons. The duplicates are identified in the scheme by comparisons that can be performed in a linear time, for instance by means of a hash table. Similarly, the scheme identifies valid credentials by testing each credential apart and this can be also performed efficiently. Thus, let N be the number of eligible voters and V the number of posted votes, our scheme has a running time $O(N+V)$. As V may be much greater than N, our scheme is linear in the number of votes.

5.6 Conclusions

We have introduced a practical scheme for remote Internet voting that satisfies the coercion-resistant property. This property takes into account realistic threats for remote voting scenarios. Our solution is inspired by the ideas of Juels, Catalano, and Jakobsson. It relies on anonymous credentials that protect voters against adversaries.

The new scheme employs credentials based on Camenisch and Lysyankaya group signature protocol. The new credentials allow the scheme to avoid comparisons of ciphertexts and consequently to achieve linear work factor. Credentials based on Boneh et al. [12] group signatures can be used in a similar way.

In principle, the credentials of our proposal cannot be revoked and novel credentials would be necessary for a new voting. However, we have presented a protocol that renders possible the revocation of credentials so that credentials can be used in multiple elections. This protocol relies on the fact that part of the credentials can be made public without compromising the security of the scheme.

We have also evaluated the new scheme based on standard security requirements for electronic voting as well as the coercion-resistant property.

Chapter 6

Future Work

This work presented voting protocols for polling station and remote Internet voting. The former schemes are voter-verifiable and employ a new concept of ballot box. The remote Internet voting protocol satisfies the coercion-resistance property and is practical for large scale voting.

The new ballot box concept anonymizes elements and outputs copies of random elements as receipts after receiving a new element. It is based on the original Farnel box. We have specified the box and its requisites. In addition, we performed simulations to verify the privacy of the first voter. Although in the simulations the voter privacy is maintained for small number of initial votes, the exact value of this parameter is not clear.

Implementing the concept of the Farnel box in a way that requires minimal trust in the mechanism or procedures is challenging. We have proposed a number of possible implementations, but other implementations could be investigated in order to reduce the trust in the components. Rivest employed the original Farnel idea to overcome the reconstruction attack in the version of the Threeballot proposed in [80]. In his scheme, a copy of a vote is made in advance and then it is exchanged by means of Farnel. A similar idea could be explored to avoid making copies inside the box as we presented. That is, a number of copies of the vote or receipt (without the scratch surface) could be made in advance and then cast into an original Farnel box. Additional mechanisms to prevent the exposure of the commitment values would be necessary, though.

The first voter-verifiable scheme presented can be easily understandable by the voters since it does not employ cryptography. However, as malicious behaviors during the tally may compromise the voting integrity, this scheme requires honest talliers. In order to reduce the trust in the talliers, we have given a new version of this scheme. The proposal employs a Prêt-à-Voter style ballot and a single cryptographic primitive. These solutions require a conventional ballot box in addition to the new Farnel concept. We have also proposed a scheme that requires just a Farnel box and has a single ballot casting procedure.

Besides these schemes, we have showed variants of the Threeballot and Randell-Ryan schemes. The new version Threeballot scheme does not leak indication of the voting results as the original one. It uses a Prêt-à-Voter style ballot and a

Farnel box. Although it employs cryptography to overcome this drawback, it does not require a verification machine to check ballots; this can be performed by the officials. We have proposed a Threeballot version that does not require the Farnel box, but it requires a more complex verification machine. This scheme, though, could be improved to avoid this machine.

The Randell-Ryan scheme variant does not require honest talliers or other techniques to prevent corruption of votes. It employs a modified ballot form and replaces one of the conventional boxes by a Farnel box. Even though the new scheme is more complex than the original version due to its new components, it ensures more security guarantees than the previous one.

The voter-verifiable schemes proposed are paper-based and employ a physical Farnel box. Most of the ideas of the box, though, could be used in an electronic scheme. For instance, a direct electronic machine could generate the initial votes and commit to them. After receiving a new vote, it could decommit some random votes as receipt to the voter.

The new scheme for remote voting fulfills the same properties and overcomes coercive attacks as the proposal of Juels, Catalano, and Jakobsson. Also, it has a linear work factor. The scheme employs special credentials of which security relies on two known assumptions. In addition, it does not require comparisons against a list of ciphertexts.

We have presented a solution in which the credentials are generated directly by the officials. The protocol can be improved to issue credentials in a distributed setting. Moreover, it considers reliable client machines. As clients could be compromised by malicious softwares in practice, the scheme can be modified to overcome these threats.

Bibliography

[1] Alessandro Acquisti. Receipt-free homomorphic elections and write-in ballots. Cryptology ePrint Archive, Report 2004/105, 2004. `http://eprint.iacr.org/`.

[2] Ben Adida. *Advances in cryptographic voting systems*. PhD thesis, Cambridge, MA, USA, 2006. Adviser-Ronald L. Rivest.

[3] Ben Adida and Ronald L. Rivest. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 29–40, New York, NY, USA, 2006. ACM.

[4] Ammar Alkassar and Melanie Volkamer, editors. *E-Voting and Identity, First International Conference, VOTE-ID 2007, Bochum, Germany, October 4-5, 2007, Revised Selected Papers*, volume 4896 of *Lecture Notes in Computer Science*. Springer, 2007.

[5] Roberto Araújo, Ricardo Felipe Custódio, and Jeroen van de Graaf. A verifiable voting protocol based on Farnel. IAVoSS Workshop On Trustworthy Elections (WOTE2007), Ottawa, Canada, June 2007.

[6] Roberto Araújo, Sébastien Foulle, and Jacques Traoré. A practical and secure coercion-resistant scheme for remote elections. In David Chaum, Miroslaw Kutylowski, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Frontiers of Electronic Voting*, number 07311 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[7] Roberto Araújo and Peter Y. A. Ryan. Improving the Farnel voting scheme. In Robert Krimmer and Ru¨diger Grimm, editors, *Electronic Voting*, volume 131 of *LNI*, pages 169–182. GI, 2008.

[8] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *PODC*, pages 274–283, 2001.

[9] Josh Daniel Cohen Benaloh. *Verifiable secret-ballot elections*. PhD thesis, 1987.

[10] Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In Alkassar and Volkamer [4], pages 111–124.

[11] Dan Boneh. The decision diffie-hellman problem. In Joe Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.

[12] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Franklin [37], pages 41–55.

[13] Stefan A. Brands. An efficient off-line electronic cash system based on the representation problem. Technical report, Amsterdam, The Netherlands, The Netherlands, 1993.

[14] Ernest F. Brickell, editor. *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*. Springer, 1993.

[15] Johannes Buchmann. *Introduction to Cryptography*. Springer, 2004.

[16] Christian Cachin, Klaus Kursawe, and Victor Shoup. Random oracles in constantipole: practical asynchronous byzantine agreement using cryptography (extended abstract). In Gil Neiger, editor, *PODC*, pages 123–132. ACM, 2000.

[17] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Franklin [37], pages 56–72.

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

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

[20] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Brickell [14], pages 89–105.

[21] David Chaum, Ron Rivest, Markus Jakobsson, Berry Schoenmakers, Peter Ryan, Josh Benaloh, and Mirek Kutylowski, editors. *Towards Trustworthy Election Systems*. 2008. To appear.

[22] David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical voter-verifiable election scheme. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 118–139. Springer, 2005.

[23] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: A secure remote voting system. Technical Report TR2007-2081, Cornell University, May 2007.

[24] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy*, pages 354–368. IEEE Computer Society, 2008.

[25] Michael R. Clarkson and Andrew C. Myers. Coercion-resistant remote voting using decryption mixes. Workshop on Frontiers in Electronic Elections, 2005.

[26] Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382. IEEE, 1985.

[27] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein. *Introduction to Algorithms.* The MIT Press, second edition, 2001.

[28] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[29] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In Walter Fumy, editor, *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 1997.

[30] Ricardo Custódio. Farnel: um protocolo de votação papel com verificabilidade parcial. Invited Talk at Simpósio Segurança em Informática (SSI), November 2001.

[31] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In Ivan Damgård, editor, *Lectures on Data Security*, volume 1561 of *Lecture Notes in Computer Science*, pages 63–86. Springer, 1998.

[32] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Coercion-resistance and receipt-freeness in electronic voting. In *CSFW*, pages 28–42. IEEE Computer Society, 2006.

[33] Estonia. The national electoral committee of Estonia, July 2008. Available at: `http://www.vvk.ee/engindex.html`.

[34] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426. ACM, 1990.

[35] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[36] California Internet Voting Task Force. A report on the feasibility of Internet voting, January 2000. Available at: `http://www.ss.ca.gov/executive/ivote`.

[37] Matthew K. Franklin, editor. *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004.

[38] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

[39] Geneva. The state of Geneva web site, July 2008. Available at: `http://www.geneve.ch/evoting/english/welcome.asp`.

[40] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In Jacques Stern, editor, *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 1999.

[41] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure applications of pedersen's distributed key generation protocol. In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2003.

[42] Oded Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.

[43] Oded Goldreich. Zero-knowledge twenty years after its invention. *Electronic Colloquium on Computational Complexity (ECCC)*, (063), 2002.

[44] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC*, pages 291–304. ACM, 1985.

[45] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[46] Andrew Gumbel. *Steal This Vote: Dirty Elections and the Rotten History of Democracy in America.* Nation Books, 2005.

[47] Joseph P. Harris. Election administration in the United States. The Brookings Institution, 1934. Available at: `http://vote.nist.gov/election_admin.htm`.

[48] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556. Springer, 2000.

[49] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2000.

[50] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In Dan Boneh, editor, *USENIX Security Symposium*, pages 339–353. USENIX, 2002.

[51] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In Ueli M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer, 1996.

[52] D. Jefferson, A. Rubin, B. Simons, and D. Wagner. A security analysis of the secure electronic registration and voting experiment, 2004.

[53] Aleks Essex Jeremy Clark and Carlisle Adams. On the security of ballot receipts in e2e voting systems. IAVoSS Workshop On Trustworthy Elections (WOTE2007), Ottawa, Canada, June 2007.

[54] Rui Joaquim and Carlos Ribeiro. Codevoting protection against automatic vote manipulation in an uncontrolled environment. In Alkassar and Volkamer [4], pages 178–188.

[55] Douglas W. Jones. A brief illustrated history of voting, 2001. Available at: `http://www.cs.uiowa.edu/~jones/voting/pictures/`.

[56] Douglas W. Jones. Chain voting, August 2005. Available at: `http://vote.nist.gov/threats/papers/ChainVoting.pdf`.

[57] Douglas W. Jones. Threats to voting systems. NIST workshop on Threats to Voting Systems, October 2005. Available at: `http://vote.nist.gov/threats/papers/threats_to_voting_systems.pdf`.

[58] Ari Juels and John G. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *NDSS*. The Internet Society, 1999.

[59] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. Cryptology ePrint Archive, Report 2002/165, 2002. Available at: `http://eprint.iacr.org/`.

[60] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In Vijay Atluri, Sabrina De Capitani di Vimercati, and Roger Dingledine, editors, *WPES*, pages 61–70. ACM, 2005.

[61] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography.* Chapman & Hall, 2007.

[62] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy*, pages 27–. IEEE Computer Society, 2004.

[63] Byoungcheon Lee. *Zero-knowledge Proofs, Digital Signature Variants, and Their Applications.* PhD thesis, School of Engineering - Information and Communications University, Daejeon, Korea, December 2001.

[64] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. Proceeding of JWISC2000, Jan 2000. Proceedings of JWISC2000, pages 101–108, Jan. 25-26, 2000, Okinawa, Japan.

[65] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In Pil Joong Lee and Chae Hoon Lim, editors, *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 389–406. Springer, 2002.

[66] Yehuda Lindell, Anna Lysyanskaya, and Tal Rabin. On the composition of authenticated byzantine agreement. *J. ACM*, 53(6):881–917, 2006.

[67] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 1999.

[68] Wenbo Mao. *Modern Cryptography Theory and Practice.* Prentice Hall International, 2004.

[69] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* 1996.

[70] Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 373–392. Springer, 2006.

[71] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security*, pages 116–125, 2001.

[72] National Institute of Standards NIST and Technology. FIPS 180-2 – Secure Hash Standard. Available at `http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf` (05.11.2005), August 2002.

[73] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Brickell [14], pages 31–53.

[74] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 248–259. Springer, 1993.

[75] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[76] Torben P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In Donald W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 522–526. Springer, 1991.

[77] Stefan Popoveniuc and Ben Hosp. An introduction to Punchscan. IAVoSS Workshop On Trustworthy Elections (WOTE'06), June 2006.

[78] Brian Randell and Peter Y. A. Ryan. Voting technologies and trust. *IEEE Security and Privacy*, 04(5):50–56, 2006.

[79] Anna Redz. On equality testing protocols and their security, September 2003. Licentiate Thesis.

[80] Ronald L. Rivest. The Threeballot voting system. Available at: `http://people.csail.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf`, October 2006.

[81] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[82] Ronald L. Rivest and Warren Smith. Three voting protocols: Threeballot, VAV, and Twin. Electronic Voting Technology Workshop (EVT), August 2007.

[83] Peter Y. A. Ryan. A variant of the Chaum voting scheme. Technical Report CS-TR-864, University of Newcastle upon Tyne, 2004.

[84] Peter Y. A. Ryan. Prêt à voter with a human-readable, paper audit trail. Technical Report CS-TR-1038, University of Newcastle upon Tyne, 2007.

[85] Peter Y. A. Ryan and Steve A. Schneider. Prêt à voter with re-encryption mixes. In Dieter Gollmann, Jan Meier, and Andrei Sabelfeld, editors, *ESORICS*, volume 4189 of *Lecture Notes in Computer Science*, pages 313–326. Springer, 2006.

[86] Steve Schneider. *Concurrent and Real Time Systems: The CSP Approach.* John Wiley & Sons, Inc., New York, NY, USA, 1999.

[87] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[88] Berry Schoenmakers. Personal communication, 2000.

[89] Jörn Schweisgut. Coercion-resistant electronic elections with observer. In Robert Krimmer, editor, *Electronic Voting*, volume 86 of *LNI*, pages 171–177. GI, 2006.

[90] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[91] Warren Smith. New cryptographic election protocol with best-known theoretical properties. Workshop on Frontiers in Electronic Elections, 2005.

[92] Charlie E. M. Strauss. A critical review of the triple ballot voting system. part 2: Cracking the triple ballot encryption, October 2006.

[93] Jeroen van de Graaf. Merging prêt à voter and punchscan. Cryptology ePrint Archive, Report 2007/269, 2007. Available at: `http://eprint.iacr.org/`.

[94] Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Commun. ACM*, 47(2):56–60, 2004.

[95] Stefan G. Weber, Roberto Araújo, and Johannes Buchmann. On coercion-resistant electronic elections with linear work. In *2nd Workshop on Dependability and Security in e-Government (DeSeGov 2007) at 2nd Int. Conference on Availability, Reliability and Security (ARES'07)*, pages 908–916. IEEE Computer Society, 2007.

[96] Z. Xia, S. Schneider, J. Heather, P. Ryan, D. Lundin, R. Peel, and P. Howard. Prêt à voter: All-in-one. IAVoSS Workshop On Trustworthy Elections (WOTE2007), Ottawa, Canada, June 2007.

[97] Zhe Xia, Steve A. Schneider, James Heather, and Jacques Traoré. Analysis, improvement, and simplification of prêt à voter with paillier encryption. Electronic Voting Technology Workshop (EVT), July 2008.

Wissenschaftlicher Werdegang

Apr. 2005	–	Sep. 2008	Doktorand am Lehrstuhl Prof. J. Buchmann, Fachbereich Informatik, Technische Universität Darmstadt
Aug. 2002	–	Apr. 2003	Stipendiat am Instituto Nacional de Pesquisas da Amazônia - Brasilien
Mar. 2001	–	Mai 2002	Studium der Informatik (M.Sc.) an der Universidade Federal de Santa Catarina - Brasilien
Feb. 1998	–	Dez. 2000	Studium der Datenverarbeitungtechnologie an der Universidade da Amazônia - Brasilien