On Coercion-Resistant Electronic Elections with Linear Work

Stefan G. Weber, Roberto Araújo, Johannes Buchmann
Darmstadt University of Technology
Department of Computer Science
Hochschulstrasse 10, 64289 Darmstadt, Germany
sweber@tk.informatik.tu-darmstadt.de, {rsa,buchmann}@cdc.informatik.tu-darmstadt.de

Abstract

Remote electronic voting over the Internet is a promising concept to afford convenience to voters and to increase election turnouts. However, before employing electronic voting systems in regular elections, problems such as coercion and vote selling have to be solved. Recently, Juels, Catalano and Jakobsson introduced a strong security requirement that deals with theses concerns. Coercion resistance improves on the former security notion of receipt freeness by taking additional real-life threats into account. In this paper, we present a coercion-resistant election scheme with a linear work factor. The scheme is based on the previous proposal of Juels et al., which exhibited a quadratic work factor, and employs Smith's idea to achieve a speedup to linear work. It, however, overcomes the drawbacks of these preceding solutions. We also present an evaluation of the scheme and identify the drawbacks and the real world aspects related to the scheme.

1 Introduction

Voting technology is imminent to take the next step towards *remote electronic voting*. As consequence, it will not only reduce the costs of conducting elections, but also afford convenience to the voters. In particular, the offered convenience will allure those groups of voters that regularly abstain from elections.

Although remote elections have many advantages, problems such as voter coercion and vote-selling have to be solved before they can be used for regular elections. While traditional elections usually employ physical mechanisms (e.g. voting booths) and supervised environments to ensure security, there are no such restrictions in the remote case. This facilitates possible abuses (coercion and vote-selling), since the voter can be watched and thus controlled while casting her ballot. Concepts, as the one implemented in the 2005 local elections in Estonia [3], can help to counter these abuses; in the Estonian case, a voter could vote multiple times over the Internet, cancelling the previously cast ballots. Other problems are cumbersome as well; it is difficult to assure that client computers are free of malicious software. A worm or trojan, for example, could be easily sent out by attackers to collect the voters' ballots. The challenging goal, thus, is the development of solutions that allow the voter to cast her vote securely in a non-supervised environment and in the presence of adversaries.

Former electronic voting schemes dealing with coercion and vote-selling impeded the voter to make receipts, i.e. from creating information that documents her vote. However, in order to implement the receipt freeness requirement, voting schemes have employed untappable channels or, alternatively, tamper-resistant hardware during the voting phase. In addition, these schemes have assumed that the voter cannot be observed by third parties while voting. Thus, since the voter is unable to construct receipts and to send them over the network, she cannot be coerced to vote for a certain candidate or sell her vote. Receipt-free schemes can be found in [12],[1],[17].

Receipt freeness is an important requirement to counter coercion and vote-selling, but it is not sufficient to eliminate these problems completely and the assumptions claimed before are not realistic for remote elections. Recently, Juels et al. [15] introduced the concept of coercion resistance. This requirement is more reasonable for remote elections. It does not only consider the receipt problem, but also a possible interaction between the adversary and the voter during the voting stage, and the revelation of private information by the voter. Furthermore, coercion resistance also considers that the voter can be forced to post a random vote encryption and that she can be forced to abstain from voting. Juels et al. [15] also proposed the first coercion-resistant election scheme that took these factors into account. To implement

coercion resistance, the scheme assumes that only the registration phase is secure, and that adversaries cannot observe the voter continuously in the voting phase. The drawback, however, is the overhead needed to compute the voting results: it is quadratic in the number of votes. Specifically, the overhead is caused by comparing non-deterministic encryptions via pairwise blind comparisons.

Aiming mainly at solving the overhead problem, Smith [25] introduced an improved scheme. His proposal replaces the inefficient comparisons by making encryptions deterministic, without leaking information about the plaintext, and then comparing them directly using hashtables. This makes it possible to compute the election result in linear time. In addition to the new method of comparison, Smith pointed out weaknesses in Juels et al.'s scheme and proposed countermeasures in his protocol.

However, the Smith's scheme is not clear in some aspects, and we find out that his method of comparison can fail in some special cases. Consequently, this problem causes the scheme to compute inaccurate election results. Moreover, the weaknesses pointed out and corrected by Smith do not seem to be relevant.

Contribution and Organization

We improve the Smith's approach for realizing a coercion-resistant election scheme with a linear work factor. The resulting scheme is basically a combination of the Juels et al. scheme with an improved version of Smith's approach. Furthermore, we present an evaluation of the enhanced scheme and discuss practical aspects related to schemes based on Juels et al.'s approach.

In the following Section, we briefly present the proposal of Juels et al., and Smith's suggestions for improvements. Moreover, we also review the drawbacks inherent in both solutions. In Section 3, we then describe the enhanced coercion-resistant scheme. An evaluation of the scheme regarding security requirements follows in Section 4. In Section 5 we present practical issues related to the implementation of election schemes based on Juels et al.'s approach. Finally, in Section 6, we present our conclusions.

2 The JCJ scheme and the Smith's improvements

In this section we briefly introduce the scheme of Juels, Catalano and Jakobsson (JCJ) [15]. Furthermore, we also describe the improvements proposed by Smith [25] and point out their problems.

2.1 The JCJ scheme

The scheme proposed by JCJ is the first one that implements the requirement of coercion resistance. The key idea of the scheme is to enable the voter to deceive an adversary (e.g. a coercer) about her true intention. To achieve this, JCJ proposed an indirect authentication and authorization mechanism using anonymous credentials. Additionally, the voter is able to vote multiple times, but just one vote is identified as valid in the tallying process.

The anonymous credential is a secret piece of data that the voter receives during the registration phase. The voter includes her credential to her vote when she wants the vote to be accounted. In order to deceive a coercer, though, the voter can issue an invalid credential and use it instead. As the coercer (or the vote-buyer) cannot distinguish between valid or invalid credentials, they cannot be sure about the voter's true intention.

In the following paragraphs we briefly describe the JCJ scheme. The protocol assumes a standard modelling in a distributed setting, the bulletin board communication model [2], and verifiable mixnets [11]. The election results, and all intermediate steps are published on public readable bulletin boards.

Registration phase. The voter receives an unique credential through an untappable channel. A probabilistic encryption of the credential is also generated and published on a bulletin board. This phase is assumed to be trustworthy, i.e. voters must receive valid credentials without the interference of adversaries.

Voting phase. The voter submits her ballot over an anonymous channel to a bulletin board. The ballot is formed by probabilistic encryptions of her vote and the credential. In addition, it also contains three non-interactive zero knowledge proofs: one to prove that the encrypted vote encodes a valid candidate, and two others to prove the voter knows the data encrypted as credential and vote.

Tallying phase. In this phase ballots are processed by

1. excluding ballots with invalid proofs;
2. eliminating ballots containing duplicate credentials via pairwise blind comparisons of encrypted credentials, keeping just the last-posted ballots; these ballots can be identified by the order of postings on the bulletin board;
3. mixing the list of ballots as well as the list of encrypted credentials from the registration phase. After the mixing process, credentials from both lists are checked

via pairwise blind comparisons. The credentials that match point out valid votes;

4. cooperatively decrypting and tallying the votes.

The pairwise blind comparisons of encrypted credentials are realized via plaintext equality tests (PET) [13]. The PET primitive allows to test whether two probabilistic encrypted ciphertexts represent the same plaintext, but without revealing the plaintext. The comparison using PETs, however, turns the scheme inefficient and impractical for large elections. The overhead to compute the results is quadratic in the number of votes; furthermore, quadratic storage and verification work factors arise.

2.2 The Smith's improvements

Smith [25] presented a variation of the JCJ scheme. The proposal differs from the JCJ scheme basically by applying timestamps to identify the last posted ballot, by adding a new mixing step in the tallying phase, and by replacing the method of comparison.

Instead of using plaintext equality tests, the new scheme employs a global blind comparison. The method is described as following: let g be an ElGamal system parameter, $h = g^s$ the election authorities' ElGamal public key, s the private key secret shared between these authorities, z a further private key secret shared between the authorities, sz the product of s and z (also secret shared between the authorities), r a random number, and $(g^r, h^r m)$ an ElGamal encryption of a message m.

1. The authorities cooperatively compute $(g^r)^{sz} = h^{rz}$ and $(h^r m)^z = h^{rz} m^z$.

2. By dividing, m^z is computed.

3. Finally the first 50% of the bits of m^s function as a deterministic fingerprint.

After the election authorities have applied this method to all encrypted credentials, they compare them (analog to steps 2, 3 of the tallying phase in Section 2.1) using hashtables [6]. The method does not leak information about the credentials. Furthermore, an attacker cannot reproduce the deterministic fingerprints, as he is not in possession of the keys s and z.

The new method of comparison is more efficient and speeds up the computation of the results to linear time. However, undesirable collisions can occur. Since just 50% of m^z is used, the image space of this mapping is smaller than the domain space. Thus, for a large number of votes, the comparison can point collisions that do not exist. Consequently, the collisions can turn valid votes invalid, or vice versa, producing an incorrect election result.

The new mixing step aims at preventing the 1009 attack. Smith presented it as a problem of the JCJ scheme. In this attack a coercer forces a voter to issue a large number of identical valid ballots (e.g. 1009) and expects to detect them after the tallying step 2, the duplicate elimination. If he perceives exactly 1009 identical ballots, Smith assumed that the coercer succeeded because one of them will be sent to the next phase and included into the final tally. Otherwise, the voter deceived the coercer. The JCJ scheme, however, is immune to this attack. The voter can act as intended by the coercer, but the coercer has no way to verify the validity of the coerced ballot in step 3. Thus, the attack does not work, and hence there is no reason for the new mixing step.

In the tallying step 2 of the JCJ scheme, the duplicate elimination, the last-cast among those ballots with the same credential is detected. Here, this can be realized according to the order of postings on the bulletin board. Differently, Smith introduced a new mixing step before the step 2 of JCJ. As the duplicates are only detected after the mixing, the initial order of postings on the bulletin board cannot be used. To avoid this problem, Smith employed timestamps. This data is added (in plaintext form) to the ballot by the voter who publishes it on the bulletin board. However, sending plaintext timestamps jointly with the ballots through the mixnet leaks information. An attacker can take advantage of the timestamps and discover the permutations used by the mixnet.

3 Coercion-Resistant Election Protocol with Linear Work

This section presents a clarified proposal of a coercion-resistant voting scheme with linear work factor. The proposal basically combines the best of the JCJ scheme with Smith's comparison approach; it overcomes, however, problems existent in these previous solutions (see Section 2 for more details).

3.1 Building Blocks

The following cryptographic primitives are employed in the voting scheme:

3.1.1 Threshold ElGamal Cryptosystem

As basis for the scheme we employ the ElGamal cryptosystem [9], over subgroups G_q of order q of the multiplicative group Z_p^*, for large primes $p = 2q + 1$. We treat the primes p, q and a primitive element g of G_q as common system parameters.

More specifically, we utilize a threshold variant, as described by Cramer et al. [8], offering robustness and distributing the trust. The private key $s \in_R Z_q$ is generated via

the distributed key generation protocol according to Pedersen [21], and consequently it is secret shared [24] among all n election authorities. The decryption is distributed among all authorities, and a minimal number of t out n authorities are necessary to perform this operation. The election authorities public key is $h = g^s \bmod p$. A message $m \in G_q$ is probabilistically encrypted by choosing $r \in_R Z_q$ and by computing $(g^r, h^r m)$.

3.1.2 Non-interactive Zero Knowledge Proofs

Non-interactive zero knowledge proofs (NIZKPs) are used to guarantee correctness of private actions in a number of places. See e.g. [7] for discussions of these standard techniques. Non-interactivity is realized by applying the Fiat-Shamir heuristic [10]. We denote a disjunctive proof that the ciphertext constitutes an encryption of a valid candidate choice a validity proof. A description of this proof can be found in [16]. Proofs of knowledge, realized by the Schnorr identification protocol [22], are used to create plaintext aware encryptions. Thus, this prevents attacks involving an attacker posting re-encryptions of ciphertexts, i.e. credentials or votes, of another voter.

3.1.3 Mixnets

This primitive, introduced by Chaum [4], is a cryptographic tool to anonymize messages. It affords anonymity basically by re-encrypting (or decrypting) and by permuting messages. In the following construction we employ ElGamal based re-encryption mixnets [20]. Moreover, we require the mixnets to be verifiable, i.e. to provide proofs of correctness of their operations, such as the proposals of Furukawa et al. [11] and Neff [18].

3.1.4 Bulletin Boards

The proposal employs bulletin boards [2]. These boards are public broadcast channels with memory to store information. We assume the bulletin boards to be append-only, i.e. once the information is sent to a board, it is stored and cannot be changed or deleted. The fact that a board is public readable enables anyone to verify its information. Thus, the bulletin boards help the scheme to be universal verifiable. See e.g. [23] for more discussions on their use.

3.2 Method of Comparison and Computation of the Fingerprints

As presented in Section 2.2, Smith's method to compare credentials can compromise the exactness of his scheme. In the next paragraphs we specify an enhanced method of comparison that is realized cooperatively by the election authorities. The method is based on the Shamir secret sharing [24] and on the Pedersen distributed key generation protocol [21]; in contrast to Smith's approach, it does not rely on multiplied shared secrets, making the protocol more easy to implement.

The method works as following: first all n election authorities jointly generate a secret shared hashkey z. After that, the authorities cooperatively apply their shares of z to an ElGamal ciphertext; this process blinds the plaintext inside the ciphertext. Then, the ciphertext is decrypted, yielding an blinded plaintext (the deterministic fingerprint). Finally, after processing all ciphertexts, they can be compared without leaking information about the plaintext by using these fingerprints.

To jointly generate the required secret shared hashkey z, the authorities employ the distributed key generation protocol due to Pedersen. In this protocol, each authority j ($j = 1 \cdots n$), receives a share z_j (of the hashkey z) and is publicly committed to its share, as a commitment value, $\rho_{z_j} = g^{z_j}$ (g is the generator of the subgroup G_q. See threshold ElGamal cryptosystem on Section 3.1), produced in the protocol run, is published.

We present the following **distributed blinding protocol** (analog to the distributed decryption protocol [8] of the ElGamal threshold cryptosystem) to blind a message $x \in G_q$ using the secret shared hashkey z. This method is used to apply z cooperatively to an ElGamal ciphertext.

1. Each authority computes $b_j = x^{z_j}$, a partial blinding of x, by applying its secret z_j; she also publishes publicly b_j together with a non-interactive zero knowledge proof that
$$\log_g \rho_{z_j} = \log_x b_j$$
; this is realized using a proof of knowledge for equality of discrete logs [5]. The proof assures that the authority indeed utilized its correct share to produce the partial blinding.

2. For any subset Λ of n authorities with valid zero-knowledge proofs, the blinded value x^z is reconstructed using the discrete Lagrange interpolation
$$x^z = \prod_{j \in \Lambda} b_j^{\lambda_{j,\Lambda}} \bmod p$$
where
$$\lambda_{j,\Lambda} = \prod_{l \in \Lambda \setminus \{j\}} \frac{l}{l - j} \bmod q$$
are the appropriate Lagrange coefficients.

Let $\sigma_i \in G_q$ be an unique credential, and $(g^r, h^r \sigma_i)$ an ElGamal encryption of σ_i with $r \in_R Z_q$, the authorities produce the deterministic fingerprint through the following steps:

1. To each component of $(g^r, h^r\sigma_i)$ the distributed blinding protocol is applied, blinding it to a fix secret shared exponent $z \in Z_q$: $((g^r)^z, (h^r\sigma_i)^z) = (g^{rz}, h^{rz}\sigma_i^z)$.

2. The blinded ciphertext is jointly decrypted to the blinded plaintext σ_i^z using the distributed decryption protocol of the threshold ElGamal cryptosystem.

Here, $h_z(\sigma_i) = \sigma_i^z$ denotes a fingerprint produced with a hashkey z.

The method described above is used to identify duplicated ballots and to authorize ballots in the voting scheme below, by means of hashtables. A hashtable is a data structure which maps $keys$ to $items$. Equivalent items are supposed to have equivalent keys, different items are supposed to have different keys. Let key be a fingerprint and $item$ a ballot, to compare fingerprints all *(key, item)* tuples are added to a public hashtable (we denote this performing a hashtable lookup). As equivalent $items$ have equivalent keys, the occurrence of collisions detects duplicate $items$, or can be used to authorize $items$, if the hashtable is prefilled with $keys$ of valid $items$.

3.3 Protocol Description

Taking into account the primitives from Section 3.1, and the method of comparison introduced in Section 3.2, we now describe the coercion-resistant election protocol. It contains an additional pre-processing step to speed up the tallying phase.

As the previous proposals of JCJ [15] and Smith [25], the protocol enables each voter to cast votes repeatedly; though, just one vote is accounted in the final tally. We denote: V_i an eligible voter $i = 1, \ldots, m$, BB_l an bulletin board $l = 0, \ldots, 12$, EAs the n elections authorities that share the private key SK_{EA} with the corresponding public key PK_{EA}, RA a trustworthy registration authority, σ_i a credential transmitted to the voter V_i, and σ_{R_i} the corresponding credential held by the RA.

Setup The election authorities (EAs) jointly agree on the appropriate system parameters p, q, g and cooperatively generate PK_{EA}, and SK_{EA} by using the threshold ElGamal cryptosystem. In addition, they also produce two secret shared hashkeys (k and j) via the distributed key generation protocol (see section 3.2). PK_{EA} as well as the ElGamal system parameters are published on the BB_0.

Registration Each voter V_i is registered by a trustworthy registration authority RA. Particularly, the RA generates an unique credential $\sigma_i \in_R G_q$ and transmits it to V_i; this process is supposed to be secure and is realized by using an untappable channel (see Okamoto [19] for a precise definition of an untappable channel). At end of this phase the RA publishes on BB_1 a list of all registered (eligible) voters and their encrypted credentials $C_{R_i} = E_{PK_{EA}}(\sigma_{R_i})$.

Pre-Processing of Credentials on Registration List The EAs send the encrypted credentials C_{R_i}, published on BB_1, through a verifiable mixnet. The output, unlinkable to BB_1, is posted on BB_2. For each re-encrypted credential on BB_2, the authorities jointly compute a deterministic fingerprint $h_j(\sigma_{R_i})$, using hashkey j, and post it on BB_3; this board contains the credential's fingerprints of the eligible (hence authorized) voters. To prepare the authorization of the ballots, the election authorities put all credential fingerprints taken from BB_3 into a public hashtable.

Candidate Slate Publication The election authorities publish an integrity protected list of possible choices, i.e. the competing candidates and their unique identifiers, and announce the beginning of the voting phase and its designated deadline.

Voting To cast a ballot for a candidate, V_i submits the tuple (B_i, P_i, C_i) to $BB4$ over an anonymous channel, i.e. a channel such that a coercer cannot determine whether or not a given voter casts a ballot. B_i and C_i are probabilistic encryptions (realized under PK_{EA}) of the vote and the credential, respectively; P_i consists of a validity proof anded with a proof of knowledge of the credential and a proof of knowledge of the vote (see Section 3.1).

Tallying In order to identify the votes on BB_4 with valid credentials and to tally them, the election authorities perform the following steps:

Verifying Proofs The EAs verify the NIZKPs associated with the ballots. The ballots with valid proofs are posted on BB_5, and the P_i datafields of them are discarded. Ballots with invalid proofs will not be processed furthermore.

Elimination of Duplicates For each ballot on BB_5, the EAs jointly compute a deterministic fingerprint $h_k(\sigma_i)$ of the encrypted credential C_i, using hashkey k. These values are posted on BB_6 together with the associated ballots. To detect duplicate ballots, i.e. ballots with the same fingerprint $h_k(\sigma_i)$, the EAs perform a hashtable lookup. If a collision is found, i.e. two identical credential fingerprints, only the last-posted ballot according to the order of postings on BB_6 is stored in the hashtable. This process retains at most one ballot per credential; only the last-cast ballots stored in the hashtable will be processed further. The final list of ballots with unique credentials is posted on BB_7.

Mixing The EAs apply a verifiable mixnet to ballots with unique credentials on BB_7. The mixnet anonymizes the remaining B_i and C_i datafields of each ballot. The output of the mixnet is posted on BB_8.

Authorizing Votes The EAs jointly compute a deterministic fingerprint for each encrypted credential associated with a ballot on BB_8, using hashkey j. The fingerprints $h_j(\sigma_i)$ are posted together with the associated ballots on BB_9. To authorize the ballots, i.e. to sort out ballots with invalid credentials, the EAs use the hashtable prepared in step 3; this table is filled with fingerprints of valid credentials issued to eligible voters. For each ballot on BB_9, a hashtable lookup on the associated fingerprint $h_j(\sigma_i)$ is performed. If a collision is found, the ballot is posted on BB_{10} as an authorized ballot.

Decrypting Votes For each authorized ballot on BB_{10}, the EAs jointly decrypt the B_i datafield, i.e. the encrypted vote, and post it on BB_{11}.

Tallying The EAs cooperatively compute the final election results from BB_{11} and publish the final tally on BB_{12}.

4 Protocol Analysis

In this section we sketch the analysis of the presented protocol. The analysis takes into account the coercion resistance requirement as well as other requirements usually implemented by voting schemes. The analysis partly holds for the original JCJ scheme as well.

4.1 Coercion Resistance

As defined by JCJ, a voting scheme is coercion-resistant if it is receipt-free and additionally prevents the randomization, the forced-abstention, and the simulation attacks. In the following, we assume that a coercer can only corrupt a minority of election authorities. The registation phase is assumed to be realized trustworthy.

The protocol is receipt-free. As the credentials are verified after the mixing process, and due to the hashtable lookup mechanism, the voter cannot obtain evidences to make receipts, because she cannot prove that her ballot contains a valid credential.

The randomization attack means that a coercer can force a voter to submit a random information as a vote. This information can be identified in the election results. The scheme prevents this attack. The invalid vote will never be decrypted, as it will be discarded in the proof checking stage (in the tallying phase). Moreover, the voter can include an invalid credential in the ballot to assure that it will be discarded.

In the forced-abstention attack the voter is forced not to vote. The scheme is also immune to this attack. The voter submits her vote via an anonymous channel, and the authentication data (encrypted credential) is hidden during the whole run of the protocol. Thus, the coercer cannot deduce if a voter actually issued a vote or not. The required anonymous channel, i.e. a channel such that an attacker cannot determine whether or not a given voter casts a ballot, can practically be achieved by using arbitrary client computers connected to the Internet, e.g. public terminals.

The simulation attack means that the coercer can force the voter to give away her valid credential. The coercer votes on behalf of the voter then. The protocol also prevents this attack. The voter can give away an invalid credential, but the coercer cannot verify if this credential is valid or not. The mixing stages, and the hashtable lookup mechanism (see Section 3.2 for details) impede the verification.

4.2 Correctness - Collision Freeness

In the Section 2.2 we argued that the Smith's scheme can produce incorrect election results. The results computed by the enhanced scheme, differently, are accurate. This is true because the credential hashing does not produce undesirable collisions.

To enable the hashtable lookup mechanism, each unique credential value $\sigma \in G_q$, σ, is blinded and decrypted to $\sigma^z \in G_q$ for a secret shared hashkey z. The modular exponentiation is injective[1], $\sigma^z \in G_q$ is unique, and so free of undesirable collisions. Furthermore no negligible information about the credential is leaking.

The number of voters, and the number of valid credentials is negligible small in comparison to the credential space G_q. Thus, casting ballots with randomly chosen credentials has only a negligible probability to succeed, if appropriate ElGamal system parameters are chosen. Hence, the protocol assures that only valid votes will by included in the final tally.

4.3 Universal Verifiability

The protocol is universal verifiable, i.e. anyone can verify that it correctly processed and tallied all valid votes. Thus, the correctness of the voting scheme is transparent to external verifiers. The protocol employs public bulletin boards that are readable by anyone. Each step of the tallying phase can be verified by either checking the broadcast NIZKPs stored on bulletin boards in case secret inputs are involved, e.g. private keys or secret permutations in the

[1] For appropriate choices of z. Using $z = 0$ the mapping is clearly not injective, for $z = 1$ the plaintext is leaking.

mixing, or by re-performing the deterministic steps. Additionally, in the setup phase, anyone can check the distributed key generation.

4.4 Robustness

The robustness of the enhanced scheme lies in its ability to tolerate attacks or failures without requiring an election to be canceled. The employed (t, n) threshold ElGamal cryptosystem as well as the proposed distributed computation of fingerprints can tolerate the failure of maximum $n - t$ election authorities.

4.5 Democracy

A voting scheme must allow only valid voters to vote and may include only one vote per voter in the final tally. The JCJ scheme as well as the enhanced scheme accept votes from anyone (valid and invalid voters) submitted to the bulletin board. In order to identify votes of valid voters, though, the scheme compares the list of encrypted credentials, obtained after removing duplicates and the mix process, with the list of encrypted valid credentials. This comparison assures that only one vote per valid voter will be accounted.

4.6 Efficiency

The hashtable lookup mechanism allows the scheme to overcomes the quadratic runtime of the JCJ scheme. The amount of storage and verification of zero knowledge proofs, broadcast by the election authorities in the tallying steps, reduces to $O(V + N)$ as well, for V votes sent by N voters. If we assume that $N \in O(V)$, the work factor of the protocol can be expressed as $O(V)$. Thus, the runtime of the coercion-resistant protocol is linear in the number of votes.

5 Practical Issues

In this Section we discuss some issues that must be considered before employing JCJ based schemes in real world elections.

Storing the Credentials

A realistic valid credential is a very long bit string. It is impractical to assume that the voter will memorize her own credential, so in real world the voter will need a way to store it. An immediate solution would be smart cards (or similar devices). However, unless a voter affected by coercion could receive another device and credential, cancelling the old one, the implemented scheme would not be coercion-resistant. In addition, the voter still would need a way to make invalid credentials. A more practical solution is to store credentials on a web server under pseudonyms and protected with passwords; each voter could possess multiple pseudonyms and use them as intended. The drawback is the additional trust in the service.

Reducing the assumptions in the Registration

The registration phase is crucial for the security of JCJ based schemes, and it is assumed to be trustworthy. Particularly, the registration authority is supposed to be trustworthy and the voter must receive her credential in a secure manner. Moreover, the adversaries are assumed to coerce voters prior to the registration. In a real world scenario, though, the coercion could happen during the registration even if the authority is trustworthy. In case the voter registers physically (e.g. by registering in a secure place), a coercer could force her to reveal the credential immediately after she receives it (e.g. after she leaves the place). On the other hand, if the registration is realized remotely (e.g. by Internet), the coercer could easily observe the voter during the process. A possible way to circumvent the problem is to allow multiple registrations. This way, the voter under coercion could register again cancelling the old credential and obtaining a new one, if the coercer forces her to obtain the credential. Naturally, the registration authority must be trustworthy and cannot collude with the coercer revealing the new registration.

Bulletin Board

Voting schemes are normally supposed to identify voters before accepting and publishing their votes on the bulletin board. In JCJ based schemes, differently, the bulletin board accepts votes from anyone and the votes of eligible voters are determined in the tally phase. Although this approach helps the scheme to implement the coercion resistance requirement, problems arise in real world scenarios. Since the bulletin board is made available to arbitrary clients, an attacker could easily post a large number of ballots to delay the computation of the final tally or to try to exhaust its storing resources. In order to prevent this attack, multiple postings from the same network address could be refused; though, as the ballots from eligible voters are only identified in the tally phase, refusing postings could at the same time impede valid ballots to be accounted. Fully automated postings on the board could be prevented by using challenge-response techniques like the CAPTCHA [26]. Moreover, as proposed by JCJ, cryptographic puzzles [14] could be

employed, in which clients are forced to do computational work before using the service. These techniques, however, do not avoid that anyone posts ballots to the board.

Implementing the Fingerprints and Hashtables

Normally hashtable data structures limit the size of the key objects to realize an efficient storage mechanism. As described, to achieve collision freeness (see Section 4.2), we propose to use the complete blinded credential as deterministic fingerprint, which leads to a less efficient hashtable implementation. To overcome this problem, the complete fingerprint can be additionally stored in the ballots' datastructure, and only a small subset of it can be used as key. Now, in case of a collision in the hashtable lookup, the actual fingerprints have to be extracted from ballots and compared, to guarantee correctness.

6 Conclusion

In this paper we proposed a coercion-resistant voting scheme with linear work factor. The scheme combines the best of the JCJ proposal with an improved implementation of Smith's comparison mechanism. Furthermore, we presented an analysis of the scheme and discussed practical aspects relative to JCJ based schemes.

The main idea of JCJ based schemes is to enable the voter to deceive an adversary about her true intention. As the previous proposals, the enhanced scheme achieves this property by allowing the voter to vote repeatedly and to issue invalid credentials. We stress that just multiple votes is not enough to prevent coercion. A voter under coercion should be able to make invalid credentials and give them away or vote multiple times using them. Otherwise, if the voter uses her valid credential and tries to vote at the end of the voting phase using it, she has no way to verify whether the coercer issued a ballot after her.

The use of anonymous credentials is a refinement of the idea implemented in the 2005 Estonian elections. The Estonian concept allowed multiple votes, but ballots previously cast were cancelled. The anonymous credentials, differently, allows the voters to cast undistinguishable invalid votes additional to the valid ones. This is a strong argument for the practicality of this approach.

As described, the scheme fulfills the coercion resistance as well as the robustness, the correctness, the democracy, and the universal verifiability requirements. Differently from receipt-free schemes, the scheme assumes an untappable channel only in the registration phase that is trustworthy; in addition, it utilizes anonymous channels during voting instead of untappable channels. Thus, as the previous proposals of Juels et al. and Smith, the scheme is more realistic for remote elections.

Although not considered in the preceding proposals, several practical aspects have to be considered before employing JCJ based schemes in real world elections. We have discussed some of them and also pointed out practical solutions to help implement the scheme.

The coercion resistance requirement introduced by Juels, Catalano and Jakobsson considers a realistic scenario for remote voting. We believe that the original JCJ proposal jointly with the improvement proposed by Smith and further improved in this paper will be very useful to realize a realistic implementation of a coercion-resistant scheme, usable for regular elections.

References

[1] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard. Practical multi-candidate election system. In *PODC: 20th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 274–283, 2001.

[2] J. C. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, Department of Computer Science, September 1987.

[3] F. Breuer and A. H. Trechsel. E-Voting in the 2005 local elections in Estonia. Technical report, Council Of Europe, 2006.

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

[5] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Advances in Cryptology - CRYPTO 92*, volume 740 of *LNCS*, pages 89–105. Springer-Verlag, 1993.

[6] T. H. Cormen, C. E. Leierson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, second edition*. MIT Press, Cambridge, Massachusetts - London, England, 2001.

[7] R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology - CRYPTO 94*, volume 839 of *LNCS*, pages 174–187. Springer-Verlag, 1994.

[8] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Advances in Cryptology - EUROCRYPT 97*, volume 1233 of *LNCS*, pages 103–118. Springer-Verlag, 1997.

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

[10] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO 86*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, August 1986.

[11] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. In *Advances in Cryptology - CRYPTO 01*, volume 2139 of *LNCS*, pages 368–387. Springer-Verlag, 2001.

[12] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology - EUROCRYPT 00*, volume 1807 of *LNCS*, pages 539–556. Springer-Verlag, 2000.

[13] M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts (extended abstract). In *ASIACRYPT 00*, volume 1976 of *LNCS*, pages 162–177. Springer-Verlag, 2000.

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

[15] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections (extended abstract). In *ACM Workshop On Privacy In The Electronic Society 2005 (WPES'05)*, pages 61–70, November 2005.

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

[17] B. Lee and K. Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In P. J. Lee and C. H. Lim, editors, *ICISC*, volume 2587 of *LNCS*, pages 389–406. Springer, 2002.

[18] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In *SIGSAC: 8th ACM Conference on Computer and Communications Security*. ACM SIGSAC, 2001.

[19] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In B. Christianson, B. Crispo, T. M. A. Lomas, and M. Roe, editors, *Security Protocols Workshop*, volume 1361 of *LNCS*, pages 25–35. Springer, 1997.

[20] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *Advances in Cryptology - EUROCRYPT 93*, volume 765 of *LNCS*, pages 248–259. Springer-Verlag, 1993.

[21] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In *Advances in Cryptology - EUROCRYPT 91*, volume 547 of *LNCS*, pages 522–526. Springer-Verlag, April 1991.

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

[23] B. Schoenmakers. Fully auditable electronic secret-ballot elections. *Xootic Magazine*, Juli 2000.

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

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

[26] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):56–60, 2004.