A Practical and Secure Coercion-Resistant Scheme for Internet Voting
(Extended Abstract)

Roberto Araújo[1], Sébastien Foulle[2], and Jacques Traoré[2]

[1] TU-Darmstadt, Hochschulstrasse 10, 64289 Darmstadt, Germany
rsa@cdc.informatik.tu-darmstadt.de
[2] Orange Labs, 42 rue des Coutures, BP 6243, 14066 Caen Cedex, France
s.foulle.ext@rd.francetelecom.com, jacques.traore@orange-ftgroup.com

Abstract. Juels, Catalano, and Jakobsson (JCJ) proposed at WPES 2005 the first voting scheme that considers real-world threats and that is more realistic for Internet elections. Their scheme, though, has a quadratic work factor and thereby is not efficient for large scale elections. Based on the work of JCJ, Smith proposed an efficient scheme that has a linear work factor. In this paper we first show that Smith's scheme is insecure. Then we present a new coercion-resistant election scheme with a linear work factor that overcomes the flaw of Smith's proposal. Our solution is based on the group signature scheme of Camenisch and Lysyanskaya (Crypto 2004).

1 Introduction

Remote electronic elections may provide many benefits to democratic societies. They may increase elections turnouts, afford convenience to the voters, and reduce costs, for instance. The risks inherent in such elections, though, can discourage its use in major political elections. The main threat is that coercion and vote-selling can be easily explored by adversaries. Remote elections, thus, must have ways to prevent or at least mitigate such problems.

Most existing proposals for remote elections rely on the receipt-freeness requirement and on assumptions to deal with coercion and vote-selling. Preventing receipts to be made, though, is not enough to counter these problems as the voter can be observed while voting, for example. Many assumptions (e.g. the voter cannot give away her private key material) are unrealistic for remote scenarios.

Recently, Juels, Catalano, and Jakobsson (JCJ) [17] introduced a more complete requirement for remote elections called coercion-resistance. This concept considers not only the receipt-freeness requirement, but also real world attacks related to coercion (and vote-selling). Coercion-resistance takes into account that an adversary can force the voter to abstain from voting, can obtain private information from the voter and vote on her behalf, or can force the voter to send a randomly formed ballot as her vote.

D. Chaum et al. (Eds.): Towards Trustworthy Elections, LNCS 6000, pp. 330–342, 2010.

Besides the coercion-resistance requirement, JCJ introduced the first scheme that fulfills it. The scheme basically mitigates coercive attacks by allowing the voter to deceive adversaries about her vote intention. It, though, requires a quadratic work factor (in number of votes) to compute the voting results and hence it is impractical for large scale elections. Particularly, the scheme relies on an inefficient blind comparison mechanism to determine the results.

Based on the JCJ solution, Smith [24] presented an efficient coercion-resistant scheme. The proposal replaces the comparison mechanism of JCJ by a new one that computes the voting results in a linear time. Also, it corrects some problems observed by Smith in JCJ scheme. Weber et al. [26], 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.

Paper Contribution and Organization

In this work we first present a weakness in Smith's mechanism of comparison that makes his scheme insecure in the sense of coercion-resistance. The problem is also relevant to the scheme of Weber et al. as it employs the ideas of Smith. We introduce a new coercion-resistant scheme with a linear work factor. The solution is based on JCJ ideas, but it has a linear work factor and does not rely on inefficient comparisons to compute the voting results.

The paper is organized as follows: in Section 2 we review the proposal of JCJ and the comparison mechanism of Smith; also, we describe the weakness of Smith's solution. After that, in Section 3, we present our proposal of coercion-resistant scheme. Then, we sketch an analysis of our proposal in Section 4. Finally, we conclude our work in Section 5.

2 The Scheme of JCJ and Smith's Comparison Mechanism

In this section we recall shortly the proposal of Juels, Catalano, and Jakobsson (JCJ) and present the mechanism of comparison proposed by Smith. Also, we show the weakness of Smith's mechanism.

2.1 The Proposal of JCJ

The scheme of Juels, Catalano, and Jakobsson [17] 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. 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 mix nets. In particular, the scheme employs a plaintext equivalence test [15]. 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 mix net. The mix net returns $L2'$. Then, the list $L1$ created during the registration phase is sent to a different mix net 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 requirement, 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.

2.2 Smith's Comparison Mechanism

Based on the JCJ proposal, Smith [24] introduced a more efficient coercion-resistant scheme. The solution substitutes the previous comparison mechanism of JCJ for a new one that computes the voting results in a linear time. Moreover, it includes a timestamp in the tuple that the voter submits and a further mix step in the tallying phase. The latter improvements, though, are irrelevant as pointed out by Weber et al. [26].

The mechanism of Smith performs a global blind comparison of ciphertexts instead of pairwise comparing ciphertexts via a plaintext equivalence test. In order to accomplish this, the method makes deterministic fingerprints from probabilistic encrypted credentials and then compares the resulting fingerprints through hash tables. The method depends on the El Gamal cryptosystem and is described as follows:

Let s be an El Gamal private key shared among the talliers and corresponding to a public key $h = g^s$, where g is a group generator, k another private key shared, ks the product of k and s also shared, and $(g^r, \sigma h^r)$ the El Gamal ciphertext of a credential σ, where r is a random number. In order to make a fingerprint from $(g^r, \sigma h^r)$, the talliers cooperatively compute $(g^r)^{ks} = h^{rk}$ and $(\sigma h^r)^k = \sigma^k h^{rk}$. Then, they divide $(\sigma^k h^{rk})$ by (h^{rk}) to obtain σ^k. The talliers now use half of the bits of σ^k as the fingerprint. This process is applied to all credential ciphertexts using the same k and ks before comparing the resulting fingerprints.

Observe that the talliers need to publish σ^k before making the fingerprint. Thus, anyone can verify the fingerprint is correct.

Weakness. Smith's comparison method is efficient. However, it is insecure. Especially, an adversary can determine whether a coerced voter gave him a valid or a fake credential [1]. In order to show this, we consider the following scenario:

Suppose an adversary forces the voter to reveal her credential σ. Now, the adversary makes two tuples, one with the encryption of σ and the other with the encryption of σ^2, and publishes them on the bulletin board. In the tallying phase, after applying Smith's method, the talliers publish σ^k and σ^{2k} on the board. Now, by squaring a copy of each element on the board, the adversary is able to test if a squared element matches an element on the board. Thus, if the two votes corresponding to σ and its square were removed by the talliers, the coercer learns that σ is an invalid credential.

3 Our Coercion-Resistant Voting Scheme

As we presented before, the scheme of JCJ is inefficient for large scale elections. Also, we showed that the comparison mechanism of Smith is insecure. We now introduce a new coercion-resistant voting scheme that employs some of the JCJ ideas and that computes voting results in a linear time.

Our solution does not rely on blind comparisons to identify valid credentials. Instead, we employ a particular mathematical structure to make the credentials and use a function to identify them apart. The structure makes hard for a coercer or a dishonest voter to forge new valid credentials, even after having seen several valid ones.

The new scheme has the following advantages: its security can be proved, it is a practical linear scheme (in the number of votes posted by the voters), one cannot link the votes of a given voter in different elections, and the generation of the credentials as well as the verification of their validity can be distributed

[1] This problem was also observed independently by Clarkson et al. [8].

among several authorities. Thus, a single corrupted authority cannot give valid credentials to an attacker or tell to a coercer whether a credential is valid or not.

3.1 Building Blocks

The scheme requires the following tools:

Bulletin Boards. In order to help achieving global verifiability, our proposal relies on bulletin boards as communication model. In this model the bulletin board performs as a public broadcast channel by receiving information and by allowing anyone to read any information received. Also, once receiving information, the board stores it and cannot delete or modify it.

Bulletin boards may be implemented via a Byzantine agreement, such as the proposal of Cachin et al. [4].

Universally Verifiable Mix Nets. In some steps of our scheme we employ mix nets to provide anonymity. This cryptographic primitive was introduced by Chaum [6] and further developed by many other authors. It performs by permuting messages, and by reencrypting or by decrypting them. The scheme requires a re-encryption mix net based on the El Gamal cryptosystem as introduced by Park et al. [22]. However, in order to reduce the trust in the mix process, the mix net should be universally verifiable. That is, after mixing messages, the mix net must prove publicly the correctness of its shuffle. The proposal of Neff [19] is an example of a universally verifiable mix net.

A Threshold Cryptosystem. Our scheme relies on a threshold version of a semantically secure cryptosystem with homomorphic property, such as Paillier [21] or El Gamal [12] under secure groups. We require here, though, the Modified El Gamal cryptosystem proposed by JCJ [17]. This variant is described as follows: let G be a cyclic group of order p where the decision Diffie-Hellman problem (see Boneh [1] for details) is hard, the public key is composed of the elements $(g_1, g_2, h = g_1^{x_1} g_2^{x_2})$ with $h, g_1, g_2 \in G$ and the corresponding private key is formed by $x_1, x_2 \in Z_p$. The Modified El Gamal ciphertext of a message $m \in G$ is $(M = g_1^s, N = g_2^s, O = mh^s)$, where $s \in Z_p$ is a random number. The message m is obtained from the ciphertext (M, N, O) by $O/(M^{x_1} N^{x_2})$. In the threshold version, the El Gamal public key and its corresponding private key are cooperatively generated by n parties; though, the private key is shared among the parties. In order to decrypt a ciphertext, a minimal number of t out n parties is necessary. See Cramer et al. [9] for a description of an El Gamal threshold cryptosystem and Gennaro et al. [13] for a secure key generation protocol.

Non-Interactive Zero-knowledge Proofs. The proposal we present below also requires several zero-knowledge proof protocols. These primitives help ensuring security in our solution. The scheme employs Schnorr signatures [23] to make ciphertexts plaintext aware (i.e. the party who makes the ciphertext should be aware of what he is encrypting) and so preventing the use of the El Gamal

malleability by adversaries; in addition, the verifier should check that the components of the ciphertexts are of order p to prevent the attacks described in [18]. The solution, moreover, requires a protocol to prove that a ciphertext contains a vote for a valid candidate. This can be accomplished, for example, through the validity proof proposed by Hirt and Sako [14]. Besides these protocols, our proposal uses the discrete log equality test owing to Chaum and Pedersen [7], a protocol for proving knowledge of a representation, such as the one proposed by Okamoto [20], and a plaintext equivalence test [15]. We employ the Fiat-Shamir heuristic [10] to convert interactive zero-knowledge proofs into non-interactive ones.

3.2 Security Model

The coercion-resistance requirement takes into account typical problems of remote votings. The scheme we introduce here aims at satisfying this requirement. 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 demands secret information from the voter. The adversary has limited computational power and is able to compromise only a small number of authorities;
- The 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. Also, we assume that the voter communicates to the registrars 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. The voters are supposed to have access to these channels and use them to post their votes;
- Voters cast their votes by means of reliable machines.

3.3 The Credentials

In the proposals of JCJ and of Smith, a valid credential is a random string. Here, differently, a credential has a mathematical structure based on the group signature scheme of Camenisch and Lysyanskaya [5].

A valid credential in our scheme has the following form: let 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 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 (see [5] for details) and on the DDH assumptions. 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.

3.4 The Scheme

Taking into account the security model and the building blocks introduced as well as the new credential, we present now our coercion-resistant voting scheme. The proposal is composed of four phases, that is, setup, registration, voting, and tallying, and 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 mix net, and for computing the voting results. They share a Modified 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 a Modified El Gamal encryption of a message m constructed with T, and $D_{\widehat{T}}[m]$ is a Modified El Gamal decryption of m.

The scheme consists of the following procedures:

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 G with prime order p where the decision Diffie-Hellman problem is hard, a random generator o of 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 Modified El Gamal threshold cryptosystem. 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 G. 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 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 [16] of well-formedness for σ. 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, c each time she votes.

Voting phase. The voter casts her ballot by sending the tuple $(E_T[C], a, E_T[a^r], E_T[a^{ry}], E_T[a^{x+rxy}], o^r, P)$ through an anonymous channel to BB, where 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 G published in the setup phase, and P is a list of non-interactive zero-knowledge proofs which ensure that the vote is well-formed. In particular, P contains a proof that the vote is for a valid candidate, proofs of knowledge of the plaintexts, and a proof that $E_T[a^r]$ and o^r contain the same r. 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 remove tuples with invalid proofs. That is, T verifies that a is in 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. 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 Modified 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 verifiable mix net 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 mix net output, where $E_T[X]'$ means a re-encryption of $E_T[X]$;
5. **Identifying valid votes.** For each tuple, R first employs its secret key y to cooperatively compute v^y. Then, R checks whether v^y and w have the same plaintext using a plaintext equivalence test. 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. Now T collaborates to decrypt the resulting ciphertext processed by R. 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" [24]. 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.

3.5 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 vote 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 knows 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 $(r, a, b = a^y, c = a^{x+rxy})$. As stated before, the element r must be transmitted via an untappable channel. However, the elements $(a, b = a^y, c = a^{x+rxy})$ 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 generates new keys (x', y') and furnish the voters with new values $(a' = a^l, b' = a'^{y'}, c' = a'^{x'+rx'y'})$, 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.

4 Analysis

The scheme presented in the previous section aims at fulfilling the coercion-resistant requirement as well as standard voting security requirements. We sketch here an analysis of our scheme based on these requirements and considering the security model introduced before.

4.1 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 proves 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 her 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 avoid the adversary performing the distinction.

4.2 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 one 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.

4.3 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 mix net 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. re-encryptions) is proved through the NIZKPs.

4.4 Efficiency

As stated 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 relies 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 bigger than N, our scheme is linear in the number of votes.

5 Conclusion

The scheme of Juels, Catalano, and Jakobsson (JCJ) considers realistic threats and is more suitable for Internet elections. Unfortunately their scheme is inefficient for large scale elections. Smith proposed an improved scheme, but his solution is not coercion-resistant as we showed.

We have introduced a practical and secure scheme that satisfies the property of coercion-resistance. Our scheme inherits some ideas from the JCJ protocol as the use of anonymous credentials. It, however, employs special credentials of which security depends on the DDH and on the LRSW assumptions; moreover, it does not rely on comparisons to identify valid credentials and is efficient for large scale elections.

The solution presented is based on the group signature scheme of Camenish and Lysyanskaya. We have a variant of our proposal that employs the protocol of Boneh et al. [2]. This variant will be presented in a forthcoming paper.

We have argued that our scheme is secure, but have not formally proved this property. We will provide a formal proof in the full version of this paper.

References

1. Boneh, D.: The decision diffie-hellman problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
2. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin [11], pp. 41–55
3. Brickell, E.F. (ed.): CRYPTO 1992. LNCS, vol. 740. Springer, Heidelberg (1993)
4. Cachin, C., Kursawe, K., Shoup, V.: Random oracles in constantipole: practical asynchronous byzantine agreement using cryptography (extended abstract). In: Neiger, G. (ed.) PODC, pp. 123–132. ACM, New York (2000)
5. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin [11], pp. 56–72
6. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM 24(2), 84–88 (1981)
7. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell [3], pp. 89–105

8. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: A secure remote voting system. Technical Report TR2007-2081, Cornell University (May 2007)
9. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)
10. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
11. Franklin, M. (ed.): CRYPTO 2004. LNCS, vol. 3152. Springer, Heidelberg (2004)
12. El Gamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
13. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. In: Stern [25], pp. 295–310
14. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000)
15. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 162–177. Springer, Heidelberg (2000)
16. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
17. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Atluri, V., De Capitani di Vimercati, S., Dingledine, R. (eds.) WPES, pp. 61–70. ACM, New York (2005)
18. Lim, C.H., Lee, P.J.: A key recovery attack on discrete log-based schemes using a prime order subgroup. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 249–263. Springer, Heidelberg (1997)
19. Andrew Neff, C.: A verifiable secret shuffle and its application to e-voting. In: ACM Conference on Computer and Communications Security, pp. 116–125 (2001)
20. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell [3], pp. 31–53
21. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern [25], pp. 223–238
22. Park, C., Itoh, K., Kurosawa, K.: Efficient anonymous channel and all/Nothing election scheme. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 248–259. Springer, Heidelberg (1994)
23. Schnorr, C.-P.: Efficient signature generation by smart cards. J. Cryptology 4(3), 161–174 (1991)
24. Smith, W.D.: New cryptographic voting scheme with best-known theoretical properties. In: Workshop on Frontiers in Electronic Elections (FEE 2005), Milan, Italy (September 2005)
25. Stern, J. (ed.): EUROCRYPT 1999. LNCS, vol. 1592. Springer, Heidelberg (1999)
26. Weber, S.G., Araújo, R., Buchmann, J.: 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 2007), pp. 908–916. IEEE Computer Society, Los Alamitos (2007)