Towards Practical and Secure Coercion-Resistant Electronic Elections

Roberto Araújo[1], Narjes Ben Rajeb[2], Riadh Robbana[3], Jacques Traoré[4], and Souheib Yousfi[5]

[1] Universidade Federal do Pará, ICEN, Faculdade de Computação, Brazil
[2] LIP2, INSAT, Tunisia
[3] LIP2, Tunisia Polytechnic School, Tunisia
[4] Orange Labs, France
[5] LIP2, ENIT, Tunisia

Abstract. Coercion-resistance is the most effective property to fight coercive attacks in Internet elections. This notion was introduced by Juels, Catalano, and Jakobsson (JCJ) at WPES 2005 together with a voting protocol that satisfies such a stringent security requirement. Unfortunately, their scheme has a quadratic complexity (the overhead for tallying authorities is quadratic in the number of votes) and would therefore not be suitable for large scale elections. Based on the work of JCJ, Schweisgut proposed a more efficient scheme. In this paper, we first show that Schweisgut's scheme is insecure. In particular, we describe an attack that allows a coercer to check whether a voter followed or not his instructions. We then present a new coercion-resistant election scheme with a linear complexity that overcomes the drawbacks of these previous proposals. Our solution relies on special *anonymous credentials* and is proven secure, in the random oracle model, under the q-Strong Diffie-Hellman and Strong Decisional Diffie-Hellman Inversion assumptions.

1 Introduction

Internet elections are far from being a consensus. On one hand, many people believe that the current technology is enough for deploying such elections in large scale. On the other hand, a number of voting researchers do not recommend them nowadays. They state that Internet elections have many intrinsic problems and that these problems must be addressed before carrying out these elections in real world scenarios. Despite of the disagreement, Estonia and the city of Geneva in Switzerland have made advances towards Internet elections. They have already developed voting systems and accomplished elections over Internet. Especially, in 2007, Estonia was the first country in the world to conduct online voting in parliamentary elections.

The success of the Internet elections in Geneva and Estonia may stimulate other countries to follow them and implement Internet voting in the near future. This may be boost by the many benefits of Internet elections over the traditional ones. Voters have the possibility to vote from any convenient place including the

S.-H. Heng, R.N. Wright, and B.-M. Goi (Eds.): CANS 2010, LNCS 6467, pp. 278–297, 2010.

comfort of their residences and offices. Also, Internet elections may be more attractive for voters and consequently increase voter turnout. Other benefits include a faster computation of the voting results and a possible reduction of costs.

These elections, however, have been criticized and discouraged by specialists as they have a number of problems. One of them is the fact that Internet elections are susceptible to coercion and vote-selling. Because voters are free for voting from any place they desire, coercers and vote buyers can easily influence them to vote for their candidates. Anyone may imagine a scenario where a vote buyer offers money to a voter and later observes her voting for his candidate. In order to reach a large number of voters, adversaries may even automatize these attacks. As stated by Jefferson et al. [21], "the Internet can facilitate large scale vote buying by allowing vote buyers to automate the process".

Although coercion and vote-selling may be difficult to hold in Internet elections, a number of voting protocols that mitigate these problems were proposed. Some of them deal with these problems through the property of receipt-freeness. That is, these schemes prevent voters from making or obtaining any evidence about their votes that could be transfered to adversaries via network.

In 2005 a more powerful property with regard to coercion and vote-selling was introduced by Juels, Catalano, and Jakobsson (JCJ) [23], though. The property, called coercion-resistance, takes into account that a voter cannot be able to make receipts as the receipt-free one. Also, it considers that the adversary may threat voters to abstain from voting, to reveal her private data, or to cast random votes. The coercion-resistance is the most effective property nowadays to fight coercion and vote-selling. In order to accomplish this notion, JCJ also introduced the first scheme that satisfies it.

Related Work on Coercion-resistance

The coercion-resistant scheme of Juels, Catalano, and Jakobsson (JCJ) first appeared in 2002 in the Cryptology ePrint Archive [22]. After improvements, it was effectively published in 2005 at WPES [23]. This scheme represents another step in the development of secure Internet voting systems. It was the first scheme to fight realistic attacks not well considered in previous solutions.

JCJ's proposal relies on anonymous credentials to overcome coercive attacks. The voter receives a valid credential (i.e. an alphanumeric string) in a secure way and uses it to cast her vote. When under coercion, the voter makes a fake credential and follows the instructions of the coercer. Later on, when alone, the voter votes again using her valid credential; this is the vote that will be counted in the tallying phase. The adversary is unable to distinguish between the valid and the fake credential. This scheme, however, suffers from an intrinsic drawback. As described in their paper "the overhead for tallying authorities is quadratic in the number of voters". Consequently, their solution is impractical for large scale elections.

Following JCJ's work, several coercion-resistant schemes were proposed. Clarkson et al. [15] presented a variant of Prêt-à-Voter scheme suitable for Internet voting and based on decryption mix nets. Mix nets are cryptographic techniques used to anonymize messages (e.g. votes). They perform this by permuting a set of messages and then by decrypting (or reencrypting) the permuted messages.

Schweisgut [29] and more recently Clarkson et al. [14] proposed schemes which mitigate the inefficiency problem of the JCJ's solution. The former scheme relies on decryption mix nets and on a tamper-resistant hardware, whereas the latter is a modified version of JCJ's proposal.

One of the most promising schemes based on JCJ's ideas was introduced by Smith [30]. He presented an efficient scheme with linear work factor. Weber et al. [32], 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. Unfortunately, the solutions of Smith and Weber et al. are not coercion-resistant as showed in [2]. The problem of Smith's scheme was also noted independently by Clarkson et al. [14].

The first practical and secure coercion-resistant scheme was given by Araújo, Foulle, and Traoré [2]. This proposal, different from the previous ones, employs special formed credentials that allows the scheme to achieve a linear work factor. It avoids the mechanism of comparisons that makes the scheme of JCJ inefficient. Unfortunately, the security of their scheme is only conjectured.

Paper Contribution and Organization

In this paper, we first show a weakness of the scheme of Schweisgut [29]. In particular, we describe an attack that allows an adversary to check whether a voter followed or not his instructions.

We then introduce a new coercion-resistant voting scheme. Our solution is practical and can be used in elections that comprehend a large number of voters. The new proposal employs some ideas similar to that presented in [2]. However, our scheme differs from the previous solution mainly in two aspects. First, we employ new anonymous credentials whose security relies on a different problem. These credentials are shorter than the credentials presented in [2] and make our proposal more efficient than the previous one. Second, while in [2] they did not prove that their scheme is coercion-resistant, we formally prove that our solution fulfills this security requirement.

This work is organized as follows: in the next section, we show that the proposal of Schweisgut is not coercion-resistant. In Section 3, we first introduce the main assumptions on which rely the security of our new voting protocol. We then present the main cryptographic building blocks used in our scheme and recall the game-based definition of coercion-resistance introduced by JCJ. We next describe our new proposal. In Section 4, we present a formal security analysis of our solution. Finally, in Section 5, we conclude this work.

2 Weaknesses on a Known Coercion-Resistant Solution

In this section we briefly describe a known coercion-resistant proposal. The scheme was given by Schweisgut [29] and aims at being more efficient than JCJ's protocol. However, we show here that Schweisgut's scheme is not coercion-resistant as claimed.

2.1 The Protocol of Schweisgut

As the original proposal of JCJ, the scheme employs anonymous credentials. These credentials identify eligible voters without revealing their identity. They also allow the voter to deceive adversaries. More specifically, the voter has a valid credential that she uses when she is not under coercion. When coerced by an adversary, the voter is able to make a fake credential and use it. As the coercer cannot distinguish between a valid and a fake credential, he cannot determine whether the voter gave him a valid credential or not.

In the scheme of Schweisgut, in particular, the voter has only two credentials. One of them is a valid one and the other is a fake one. Both credentials are stored in an observer, i.e. a tamper resistant device. Taking into account that a public generator g (among other public parameters) and that an El Gamal key pair [19] of the talliers (where T is the public key) were previously generated, the scheme is briefly described as follows:

Registration Phase. After authenticating the voter, the registration authorities (registrars) generate a random valid credential σ and encrypts it producing $E_T[\sigma]$ (where $E_X[m]$ means an El Gamal encryption of a message m computed with the public key X). They then transfer $E_T[\sigma]$ to the voter that stores it in her observer. The voter now generates a random fake credential σ', encrypts it, and stores $E_T[\sigma']$ on her observer. At the end of this phase, the registrars send a list of encrypted valid credentials through a verifiable decryption mix net (i.e. the mixes have to prove that they have correctly permuted and decrypted the tuples) and publish the mix net results.

Voting Phase. In order to vote, the voter interacts with her observer. During this, she selects two random numbers a, a', uses a to encrypt her vote v and obtains $E_T[v]$; she then employs the other random number to compute $g^{a'}$, and sends $E_T[v]$ and $g^{a'}$ to the observer. The observer now selects two fresh random numbers b and b', reencrypts $E_T[v]$ with b and obtains $E_T[v]'$, and reencrypts the encrypted valid credential $E_T[\sigma]$ (or the encrypted fake credential $E_T[\sigma']$ according to the voter intention) to obtains $E_T[\sigma]'$; it then computes $g^{a'+b'}$ and $O = [b \cdot H(g, E_T[v]', E_T[\sigma]', g^{a'+b'}) + b']$, where H is a secure hash function, and sends back to the voter: $\langle g^{a'+b'}, E_T[v]', E_T[\sigma]', O\rangle$; O is a non-malleability proof. After receiving the values from the observer, the voter computes $O' = [(a + b) \cdot H(g, E_T[v]', E_T[\sigma]', g^{a'+b'}) + (a' + b')]$ and publishes on a bulletin board the following tuple: $\langle g^{a'+b'}, E_T[v]', E_T[\sigma]', O', P\rangle$, where P is a proof that $E_T[v]'$ contains a valid vote. This is performed via an anonymous channel.

Tallying Phase. Once the voting period is finished, the talliers first verify the proof of non-malleability O' and the proof P. After excluding votes with invalid proofs, the talliers apply a plaintext equivalence test[1] [20] to identify credentials used more than once (i.e. duplicates); the talliers keeps one of the duplicates based on a policy (e.g. the last posted vote). Then, the talliers send the votes (i.e. the remaining tuples $\langle E_T[v]', E_T[\sigma]' \rangle$) through a *verifiable* decryption mix net. Finally, the resulting mixed credentials are compared with the valid credentials processed in the registration phase. A match identify a valid vote. Observe that the plaintext pairs (i.e. vote and credential) are published on a bulletin board so that anyone can verify the correctness of the protocol.

2.2 A Weakness in Schweisgut's Scheme

At first glance, the scheme of Schweisgut seems to be coercion-resistant. However, the coercer could use a simple strategy to test whether a credential is valid or not. Suppose a coercer forces the voter to reveal the encrypted credential $E_T[\sigma]$ he received from the registrars. The coercer then employs this encrypted credential to compute a new ciphertext in such a way that their underlying plaintexts satisfy a specific relation R; for example, the coercer could select a random value t and compute $E_T[t \cdot \sigma]'$ from $E_T[\sigma]$ and t (by exploiting the fact that El Gamal is malleable). He then computes the proof O' for the new encrypted credential $t \cdot \sigma$. Note that the coercer can make this proof himself without needing an observer and without knowing the plaintext σ. As the proof only involves the exponent used to encrypt the vote and the coercer makes this ciphertext, he knows the corresponding exponent and can make the proof himself. For this, he selects new fresh random numbers i, j, i', j' and computes: $O' = [(i + j) \cdot H(g, E_T[v], E_T[t \cdot \sigma]', g^{i'+j'}) + (i' + j')]$; this proof will hold as the one generated by an observer since its verification is true. The coercer then posts two votes (i.e. two tuples) on the bulletin board: one with the encrypted credential $E_T[\sigma]$ received from the voter and one containing the encrypted credential $E_T[t \cdot \sigma]'$.

In the tallying phase, after sending all tuples $\langle E_T[v]', E_T[\sigma]' \rangle$ to the decryption mix net, the talliers obtain a list L with pairs $\langle v, \sigma \rangle$; this list is published on a bulletin board. In order to verify the voter gave him a valid or an invalid credential, the coercer reads a σ in L and uses the value t to compute $t \cdot \sigma$. The coercer then search on L for a value $t \cdot \sigma$. When a match is found, the coercer verifies whether the vote corresponding to the credential σ was removed or not from the count. If this occurs, the coercer learns that the voter gave him an invalid credential and punishes her. Otherwise, the coercer can be sure that he received the correct credential and rewards the voter. If no match is found, the coercer repeats the process with another credential. Observe that, in the worst case, the complexity of this attack is roughly in $O(|L|log|L|)$ operations.

[1] This is a cryptographic primitive that operates on ciphertexts in a threshold cryptosystem. The input to a Plaintext Equivalence Test is a pair of ciphertexts; the output is a single bit indicating whether the corresponding plaintexts are equal or not.

3 Our New Coercion-Resistant Protocol for Internet Voting

As we showed in Section 2, the proposal of Schweisgut is not coercion-resistant as an adversary is able to distinguish between a valid and an invalid credential. In this section we introduce a new coercion-resistant scheme. Our proposal is based on JCJ's ideas and employs new credentials that prevent adversaries from checking them. The new credentials have their security based on known problems and are different from the credentials used in past coercion-resistant proposals.

3.1 Preliminaries

Notation. Let A be an algorithm. By $A(\cdot)$ we denote that A has one input (resp., by $A(\cdot, ..., \cdot)$ we denote that A has several inputs). By $y \leftarrow A(x)$, we denote that y was obtained by running A on input x. If A is deterministic, then y is unique; if A is probabilistic, then y is a random variable. If S is a finite set, then $y \leftarrow S$ denotes that y was chosen from S uniformly at random.

By $A^O(\cdot)$, we denote a Turing machine that makes queries to an oracle O.

Let b be a boolean function. By $(y \leftarrow A(x) : b(y))$, we denote the event that $b(y) = 1$ after y was generated by running A on input x. The statement $\Pr[\{x_i \leftarrow A_i(y_i)\}_{1 \leq i \leq n} : b(x_n)] = \alpha$ means that the probability that $b(x_n) = 1$ after the value x_n was obtained by running algorithms A_1,..., A_n on inputs y_1,..., y_n is α, where the probability is over the random choices of the probabilistic algorithms involved.

According to the standard definition, we say that a quantity $f(k)$ is *negligible* in k if for every positive integer c there is some l_c such that $f(k) < k^{-c}$ for $k > l_c$. In most cases, we use the term negligible alone to mean negligible with respect to the full set of relevant security parameters. Similarly, in saying that an algorithm has *polynomial running time*, we mean that its running time is asymptotically bounded by some polynomial in the relevant security parameters.

Complexity Assumptions. The security of our voting protocol relies on the following assumptions:

In [4], Boneh and Boyen introduced a new computational problem in *bilinear* context. However, for our purpose, we will consider this problem in the classical discrete log setting, i.e. without *bilinear map*.

***q*-Strong Diffie-Hellman assumption I (*q*-SDH-I) [4]:** Let k denotes a security parameter. Let G be a group of prime order p with $2^k < p < 2^{k+1}$and g a random generator in G. We say that the q-SDH-I assumption holds in G if for all polynomial-time adversaries $\mathcal{A}$ the advantage

$$\mathrm{Adv}_{G,\mathcal{A}}^{q\text{-SDH-I}}(k) = \Pr[y \leftarrow Z_p^*; (c, A) \leftarrow \mathcal{A}(g, g^y, ..., g^{y^q}) : c \in Z_p \wedge A = g^{1/(y+c)}]$$

is a negligible function in k.

q-Strong Diffie-Hellman assumption II (q-SDH-II): Let k denotes a security parameter. Let G be a group of prime order p with $2^k < p < 2^{k+1}$and g_1 and g_2 two random generators in G. We say that the q-SDH-II assumption holds in G if for all polynomial-time adversaries $\mathcal{A}$ the advantage

$$\mathrm{Adv}^{q\text{-SDH-II}}_{G,\mathcal{A}}(k) = \Pr[y \leftarrow Z_p^*; \{(x_i, r_i) \leftarrow Z_p^2; A_i = (g_1 g_2^{x_i})^{1/(y+r_i)}; B_i = (x_i, r_i, A_i)\}_{1\leq i\leq q-1}; B = (x, r, A) \leftarrow \mathcal{A}(g_1, g_2, g_2^y, B_1, ..., B_{q-1}) : (x, r) \in Z_p^2 \wedge A = (g_1 g_2^x)^{1/(y+r)} \wedge \{B \neq B_i\}_{1\leq i\leq q-1}]$$

is a negligible function in k.

Lemma 1. *If the q-SDH-I assumption holds in G then the q-SDH-II assumption holds in G*

Proof. see [17] for a proof of this Lemma.

The security of our voting protocol also relies on the Decision Diffie-Hellman assumption and on strongest variants of this assumption.

Decision Diffie-Hellman assumption (DDH) [3]: Let k denotes a security parameter. Let G be a group of prime order p with $2^k < p < 2^{k+1}$. We let $\mathrm{D_{DDH}}$ be the distribution (g, g^x, g^y, g^{xy}) in G^4 where g is a random generator in G and x, y are uniform in Z_p. We let $\mathrm{R_{DDH}}$ be the distribution (g, g^x, g^y, g^z) where g is a random generator in G and x, y, z are uniform in Z_p subject to $z \neq xy$. We say that the DDH assumption holds in G if for all polynomial-time adversaries $\mathcal{A}$ the advantage

$$\mathrm{Adv}^{\mathrm{DDH}}_{G,\mathcal{A}}(k) = |\Pr[x \leftarrow \mathrm{D_{DDH}} \ : \mathcal{A}(x) = 1] - \Pr[x \leftarrow \mathrm{R_{DDH}} \ : \mathcal{A}(x) = 1]|$$

is a negligible function in k.

The following assumption has been introduced by Camenisch et al. [8] in order to prove the security of their e-token system.

Strong Decision Diffie-Hellman Inversion assumption I (SDDHI-I) [8]: Let k denotes a security parameter. Let G be a group of prime order p with $2^k < p < 2^{k+1}$ and g a random generator in G. Let $O_a(\cdot)$ be an oracle that, on input $z \in Z_p^*$, outputs $g^{1/(a+z)}$. We say that the SDDHI-I assumption holds in G if for all polynomial-time adversaries $\mathcal{A}$, that do not query the oracle on r, the advantage

$$\mathrm{Adv}^{\mathrm{SDDHI\text{-}I}}_{G,\mathcal{A}}(k) = |\Pr[a \leftarrow Z_q^*; \ (r, \alpha) \leftarrow \mathcal{A}^{O_a}(g, g^a); y_0 = g^{1/(a+r)}; \ y_1 \leftarrow G; \ b \leftarrow \{0, 1\}; \ b' \leftarrow \mathcal{A}^{O_a}(y_b, \alpha) : b = b'] - 1/2|$$

is a negligible function in k.

Strong Decision Diffie-Hellman Inversion assumption II (SDDHI-II): Let k denotes a security parameter. Let G be a group of prime order p with $2^k < p < 2^{k+1}$ and g_1 and g_2 two random generators in G. Let $O_a(\cdot)$ be an oracle that, on input $z, t \in Z_p^*$, outputs $(g_1 g_2^t)^{1/(a+z)}$. We say that the SDDHI-II assumption holds in G if for all polynomial-time adversaries $\mathcal{A}$, that do not query the oracle on r, the advantage

$$\mathrm{Adv}_{G,\mathcal{A}}^{\text{SDDHI-II}}(k) = |\Pr[a \leftarrow Z_q^*;\ (x, r, \alpha) \leftarrow \mathcal{A}^{O_a}(g_1, g_2, g_2^a); y_0 = (g_1 g_2^x)^{1/(a+r)};$$
$$y_1 \leftarrow G;\ b \leftarrow \{0,1\};\ b' \leftarrow \mathcal{A}^{O_a}(y_b, \alpha) : b = b'] - 1/2|$$

is a negligible function in k.

Lemma 2. *If the SDDHI-I assumption holds in G then the DDH assumption holds in G*

Lemma 3. *If the SDDHI-I assumption holds in G then the SDDHI-II assumption holds in G*

For ease of presentation, we will call in the sequel q-SDH (respectively SDDHI) assumption the q-SDH-II (respectively SDDHI-II) assumption.

3.2 Cryptographic Building Blocks

The new voting scheme requires a set of cryptographic primitives to ensure its security. We describe next these primitives.

Bulletin Boards. The new scheme requires information to be publicly published so that anyone can verify them. In order to perform this, the scheme relies on a bulletin board communication model. By using this model, the scheme allows anyone to post information on bulletin boards. However, no one can delete or alter any information published on the board. The proposal of Cachin et al. [7] may be used to implement the bulletin boards required here.

A Threshold Cryptosystem. Our scheme relies on a threshold version of a semantically secure cryptosystem with homomorphic property. We require here, though, the Modified El Gamal cryptosystem proposed by JCJ [23]. This variant, is described as follows: let G be a cyclic group of order p where the Decision Diffie-Hellman problem (see Boneh [3] 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 $g_1, g_2 \in G$ 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 of n parties is necessary. The Modified El Gamal cryptosystem is *semantically secure*

under the DDH assumption. Borrowing freely from the exposition in [23], we provide, for completeness, a sketched version of this proof. Suppose there exists a probabilistic polynomial time algorithm $\mathcal{A}$ which can break the semantic security of the Modified El Gamal cryptosystem then there exists an algorithm $\mathcal{B}$ that breaks the Decision Diffie-Hellman problem. We prove this claim by constructing $\mathcal{B}$ as follows. So assume that $\mathcal{B}$ receives on input a quadruple (g_1, g_2, h_1, h_2) from the challenger $\mathcal{C}$ of the DDH problem and has to determine whether this quadruple follows the $\mathrm{D_{DDH}}$ distribution or not. $\mathcal{B}$ constructs the public key for the Modified El Gamal scheme as follows. It chooses x_1 and x_2 at random, sets $h = g_1^{x_1} g_2^{x_2}$ and sends $(g_{1,} g_2, h)$ to $\mathcal{A}$ as the challenge parameters of the Modified El Gamal scheme.

When $\mathcal{A}$ will come up with the two messages m_0, m_1 he wants to be challenged on, $\mathcal{B}$ will proceed as follows. It flips a random (private) bit b, and encrypts m_b as follows: $(h_1^k, h_2^k, m h_1^{k x_1} h_2^{k x_2})$ where k is a random value.

Note that if the given quadruple is a DH one then the ciphertext has the right distribution. This is because $h_1^k = g_1^{k'}$ and $h_2^k = g_2^{k'}$ for some k' and $(h_1^{x_1} h_2^{x_2})^k = h^{k'}$ (for the same k').

If on the other hand, the given quadruple is not a DH one then it is easy to check that $\mathcal{A}$ gains no information at all about the encrypted message (this is because this time to decrypt, $\mathcal{A}$ has to know the secret exponents x_1 and x_2 which remains information theoretically hidden by h). The latter property will be important in the proof that our voting protocol is coercion-resistant.

Universally Verifiable Mix Nets. In some steps of our scheme we employ mix nets to provide anonymity. This cryptographic primitive was introduced by Chaum [12] and further developed by many other authors. It performs by permuting messages, and by reencrypting or by decrypting them. Our scheme requires a re-encryption mix net based on the El Gamal cryptosystem. 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 proposals of Neff [26], and Furukawa and Sako [18] are examples of universally verifiable mix nets.

Non-Interactive Zero-knowledge Proofs. The proposal we present below also requires several zero-knowledge proofs of knowledge. Zero-knowledge proofs of knowledge are interactive protocols between a verifier and a prover allowing a prover to assure the verifier his knowledge of a secret, without any leakage on it. These primitives help ensuring security in our solution. Our scheme employs a proof of knowledge of a discrete logarithm [28] 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 [9]. The solution, moreover, requires a protocol to prove that a ciphertext contains a vote for a valid candidate. Besides these protocols, our proposal uses the discrete logarithm

equality test owing to Chaum and Pedersen [13], a protocol for proving knowledge of a representation, such as the one proposed by Okamoto [27], and a plaintext equivalence test [20].

Especially, our scheme requires a zero-knowledge proof of knowledge of the plaintext related to a M-El Gamal ciphertext ($M = g_1^s, N = g_2^s, O = mh^s$), and a proof that this plaintext is $\neq 1$. The former proof is accomplished by first proving the knowledge of the discrete logarithms of the M-El Gamal terms M, N in the bases g_1, g_2, respectively. Then, we prove the representation of O in the bases B and h, where B denote the basis of a special formed plaintext m and h is the M-El Gamal public parameter. We finally prove that the discrete logarithm of M, N in the bases g_1, g_2 is equal to the second component of the representation of O in the bases B and h. A description of a similar proof can be found in [1]. For the latter proof, we prove that the discrete logarithm of M, N in the base g_1, g_2 is different from the discrete logarithm of O in the base h. This proof can be performed by means of the protocol of Camenisch and Shoup (see Section 5 of [10]).

These interactive proofs can also be used non-interactively (a.k.a *signatures of knowledge*) by using the Fiat-Shamir heuristic [16]. We will use in the sequel the following notation $\textsc{Pok}[\alpha, \beta, ... : Predicate]$ to denote a non-interactive zero-knowledge proof (NIZKP) proving that the prover knows values $(\alpha, \beta, ...)$ satisfying the predicate *Predicate*. In this notation, the Greek letters will denote the secret knowledge and the others letters will denote public parameters between the prover and the verifier. For example, using this notation, $\textsc{Pok}[\alpha : h = g^\alpha]$ will denote a proof of knowledge of the discrete logarithm of h in the base g.

3.3 Attack Model

Our coercion-resistant proposal follows the general idea presented by JCJ in their scheme. This let our scheme inherits some caracteristics from the original proposal. The security model under which our scheme relies on is similar to that one of JCJ. We take into account the following assumptions:

Limited Computational Power and Small Number of Authorities. An adversary has limited computational power and may compromise only a small number of authorities. He can force the voter to reveal any secret information that she is holding. Also, he can force her to abstain from voting or to post a random composed ballot as her vote;

Interactions with the Voter. The adversary cannot monitor or interact with the voter constantly during the whole voting process. However, he may interact with the voter occasionally during the voting;

A Registration Phase Free of Adversaries. The registration official is trustworthy and voters receive private data securely. Also, we assume that the voters communicate with the registrar via an untappable channel and without the interference of adversaries. This channel provides information-theoretical secrecy to the communication;

Anonymous Channels in the Voting Phase. The existence of 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 computers in public places to achieve this or a mix net;

Trustworthy Voting Computers. The computers that the voters use for vote are trustworthy. 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.

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.

3.4 Formal Definitions

We will use the security model introduced by JCJ [23]. The essential properties are *correctness*, *verifiability*, and *coercion-resistance*, respectively abbreviated *corr*, *ver*, and *c-resist* in the sequel. Following [23], we will only focus on the formal security definition of the property of coercion-resistance as the two other properties (correctness and verifiability) are more classical and of less relevance to our work (see JCJ for formal definitions of these two properties).

In [23] coercion resistance centers on a kind of game between the adversary $\mathcal{A}$ and a voter targeted by the adversary for coercive attack. A coin is flipped; the outcome is represented by a bit b. If $b = 0$ then the coerced voter V_0 casts a ballot of its choice β, and provides the adversary with a false voting key (fake credential) $\widetilde{sk}$; in other words, the voter attempts to evade adversarial coercion. If $b = 1$, then the voter submits to the coercion of the adversary; she gives him her valid voting key (credential) sk and does not cast a ballot. The task of the adversary is to guess the value of the coin b, that is to determine the behavior of the voter. An election scheme ES is coercion-resistant, according to JCJ's definition [23], if for any polynomially-bounded adversary $\mathcal{A}$, any parameters n and n_C, and any probability distribution D_{n,n_C}, the quantity

$$\mathrm{Adv}_{ES,\mathcal{A}}^{c\text{-}resist} = \left|\mathbf{Succ}_{ES,\mathcal{A}}^{c\text{-}resist}(\cdot) - \mathbf{Succ}_{ES,\mathcal{A}}^{c\text{-}resist\text{-}ideal}(\cdot)\right|$$

is negligible in all security parameters for any voter function V_0. Where:

- n denotes the number of voters outside the control of the adversary
- n_C denotes the total number of candidates.
- D_{n,n_C} denotes a probability distribution that models the state of knowledge of the adversary about the intentions of honest voters
- $\mathbf{Exp}_{ES,\mathcal{A}}^{c\text{-}resist}$ represents the game between the adversary and the voter.
- $\mathbf{Exp}_{ES,\mathcal{A}}^{c\text{-}resist\text{-}ideal}$ represents an *ideal* voting experiment, between the adversary and the voter, in which the adversary never sees the bulletin board.
- $\mathbf{Succ}_{ES,\mathcal{A}}^{E}(\cdot) = \Pr[\mathbf{Exp}_{ES,\mathcal{A}}^{E}(\cdot) = \text{'1'}]$

Intuitively, the definition of JCJ [23], means that in a real protocol execution, the adversary effectively learns nothing more than the election tally X. The adversary cannot learn any significant information from the protocol execution itself, even when mounting an active attack (see [23] for more details about this definition and [25,31,11] for alternative definitions in the simulation-based model and also [24] for an alternative definition in the game-based model).

3.5 Anonymous Credentials

Anonymous credentials have an important role in coercion-resistant schemes. They make possible voters to deceive adversaries when under coercion and to vote later on. In most of the coercion resistant schemes such as JCJ, a valid credential is a random string. Our scheme, however, employs a different technique of credentials that differs from past proposals. Our new credentials bear some similarities with the *membership certificates* of the group signature scheme of Boneh, Boyen, and Shacham [5].

The credentials used in our solution are presented as follows: let G be a cyclic group with prime order p where the Decision Diffie-Hellman (DDH) problem (see [3] for details) is assumed to be hard, y a secret key, (g_1, g_3) two random generators of G and (r, x) two random numbers in Z_p^*. The credential is composed of (A, r, x), where $A = (g_1 g_3^x)^{\frac{1}{y+r}}$. A credential in our system therefore corresponds to a *membership certificate* in Boneh et al's group signature scheme (we could in fact either use the original version of their membership certificates or the extended one described in Section 8 of [5]).

Due to the mathematical structure of our credentials, their security depends on two known assumptions: the *q-Strong Diffie-Hellman* and the *Strong Decisional Diffie-Hellman Inversion*. The *q-Strong Diffie-Hellman* assumption (*q*-SDH for short) ensures that even if an adversary has many genuine credentials (A_i, r_i, x_i), it is hard for him to forge a new and valid credential (A, r, x) with $(r, x) \neq (r_i, x_i)$ for all i (see Lemma 1). This assumption is known to hold for generic groups and the security of Boneh et al.'s group signature scheme also relies on it. The Strong Decisional Diffie-Hellman Inversion assumption (SDDHI), which also holds in generic group, ensures that an *active* coercer, ignoring the secret key y, cannot decide whether a triplet (A, r, x) is a valid credential or not; in other words, whether it satisfies or not the following equation: $A^{y+r} \stackrel{?}{=} g_1 g_3^x$. This way, a voter under coercion will generate a random value x' and give to the coercer a fake credential (A, r, x') instead of his valid credential (A, r, x). Under the SDDHI assumption, the adversary will not be able to distinguish between a fake credential and a valid one.

In a real-world scenario, our credential can be seen as containing two parts: a short one, that is x, which must be kept secret, and a longer one, that is (A, r). The first part (i.e. x) 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 x down. The other part (A, r) can be stored in a device or be even sent by email to the voter without compromising (under the SDDHI assumption) the credential security.

3.6 An Overview of the Scheme

Before showing the details of our proposal, we give an intuition of the new scheme.

The protocol begins in a setup phase. In this phase, a set of authorities in cooperation generate the key materials and publish the corresponding public parameters on a bulletin board. In particular, they publish the public key of a threshold homomorphic cryptosystem. After this stage, the registration phase takes place. In order to register to vote, voters prove their identities to trustworthy registration authorities (registrars). These authorities issue for each voter a unique and valid credential. The voter uses this credential to cast her vote in the voting phase.

At time of voting, the voter makes a tuple containing her vote. The tuple contains the ciphertext of the vote and ciphertexts corresponding to a credential along with a set of non-interactive zero-knowledge proofs showing the validity of these encryptions. The voter casts his vote by sending this tuple to a bulletin board via an anonymous channel. When she wants to cast her vote, the voter makes the tuple using the credential she received in the registration phase. However, if the voter is under coercion, she makes a fake credential and may either use it to cast an invalid vote (i.e. a vote that will not be counted) or give this fake credential to an adversary. The adversary is not able to distinguish between the valid and the fake credential. The voter may vote again later on using her valid credential.

In the tallying phase, a set of talliers cooperate to compute the voting results. For this, they first verify some proofs on each tuple and discard tuples with invalid proofs. After that, the talliers check part of each tuple to detect tuples posted with the same credential (duplicates). Based on the order of postings of these tuples, the talliers keep the last posted tuple and eliminate the other ones. Now, the talliers send the remaining tuples through a verifiable mix net. From the mix net output, they begin to identify the tuples posted with valid credentials. These tuples contains the votes to be counted. The validity of the credentials is checked under the encryption by exploiting the homomorphic property of the underlying cryptosystem. After identifying the tuples with valid credentials, the authorities in cooperation decrypt the corresponding vote ciphertext and publish the voting results.

3.7 The Protocol in Details

We present now the description of our scheme in details. This description consider the building blocks and the attack model presented above.

Participants and Notation. The solution is composed of four phases: setup, registration, voting, and tallying. The setup phase is where the parameters of the voting are generated. The key pairs used in the scheme, for example, are generated in this phase. In the registration phase, the voter registers to the authorities and receives a valid credential. Afterwards, in the voting phase, the voter uses her credential to express her vote intention. Finally, in the tallying

phase, the voting results are computed and published. In order to perform the steps to be described in these phases, the scheme considers three participants:

- The voter is identified by $\mathcal{B}$. She obtains a valid credential σ and is able to produce a fake credential σ'. The valid credential is used for posting a valid vote, whereas the fake one is used to deceive adversaries;
- The talliers are composed of a set of authorities and are denoted by T. They control the bulletin board, run the mix net, and compute the voting results. They share an M-El Gamal private key $\widehat{T}$ corresponding to a public key T;
- The registrars, as the talliers, are composed of more than one and are identified by R. They authenticate each eligible voter in the registration phase and issue a valid credential for her. They share a M-El Gamal private key $\widehat{R}$ associated to a public key R.

In addition to the notation above, we use the following one: BB is a bulletin board, $E_T[m]$ is a M-El Gamal encryption of a message m computed with the public key T, and $D_{\hat{T}}[m]$ is a M-El Gamal decryption of m with the private key $\widehat{T}$.

Setup of the Voting Parameters. This phase takes place prior to the registration and is necessary for the definition of the voting parameters. In order to establish these parameters, first a cyclic group $\mathbb{G}$ with prime order p is defined. The Decision Diffie-Hellman problem must be hard in this group. After that, the authorities produces four generators $g_1, g_2, g_3, o \in \mathbb{G}$. The talliers T now collaborate to generate the public key $T = (g_1, g_2, h = g_1^{x_1} g_2^{x_2})$ of the Modified El Gamal threshold cryptosystem and its corresponding private key $\widehat{T} = (x_1, x_2)$. The resulting key $\widehat{T}$ is not known by the authorities individually. Each authority knows only a share of this key. The registrars R also cooperate to establish a public key $R = g_3^y$ and the corresponding shared private key $\widehat{R} = y$.

Registration Phase. After generating their keys, the registrars are ready to issue credentials for the voters. In order to receive a secret credential, the voter first proves to the registrars that she is eligible to vote. R then selects two random numbers $r, x \in \mathbb{Z}_p$ and computes: $A = (g_1 g_3^x)^{\frac{1}{y+r}}$ (which implies that $A^{y+r} = g_1 g_3^x$ and then that $A^y = g_1 g_3^x A^{-r}$). After that, R issues to the voter the secret credential $\sigma = (A, r, x)$.[2] The registrars might generate the credential in a threshold fashion. The communication between the voter and registars is performed through an untappable channel. Following JCJ, we assume that the majority of players in R are honest and can thus ensure that R provides the voter $\mathcal{B}$ with a valid credential. Nonetheless, it is possible for R to furnish the voter with a proof that $\sigma = (A, r, x)$ is a valid credential. To do this, R has to compute a non-interactive proof of knowledge that the discrete logarithm of $g_1 g_3^x A^{-r}$ (which should be equal to A^y if $\sigma = (A, r, x)$ is a valid credential) in the base A is equal to the discrete logarithm of $R = g_3^y$ in the base g_3. In order to prevent the

[2] In a variant, the value x could be jointly generated by the voter and the registrars.

voter from transferring this proof (and thus to prevent coercion), R should instead issue a *designated verifier proof* of the equality of these discrete logarithms.

Voting Phase. In order to vote, the voter with credential (A, r, x) first selects a random $s \in \mathbb{Z}_p^*$ and computes $B = A^s$ using the element A of her credential. After that, she computes the tuple: $\langle\ E_T[v], B, E_T[B^{s^{-1}}],\ E_T[B^{rs^{-1}}],\ E_T[g_3^x], o^x, \Pi\ \rangle$ which is equal to $\langle E_T[v], B, E_T[A], E_T[A^r], E_T[g_3^x]\ , o^x, \Pi\rangle = \langle C_1, B, C_2, C_3, C_4, o^x, \Pi\rangle$. The voter then publishes his tuple on a public bulletin board by means of an anonymous channel.

The tuple is composed of the ciphertext $E_T[v]$ that contains the voter's choice, the value B, the ciphertexts $\langle E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}]\rangle$ that correspond to part of the credential σ, the ciphertext $E_T[g_3^x]$ that encrypts the public generator g_3 to the power of the part x of σ. In addition, it has a set of non-interactive zero-knowledge proofs Π. This set contains:

- (Π_1) A proof that $C_1 = E_T[v]$ encrypts a valid vote (i.e, that v represents a valid candidate choice).
- (Π_2) A proof that the voter knows the plaintext related to the ciphertext $C_2 = E_T[B^{s^{-1}}] = (M_1, N_1, O_1)$. In particular, the voter will have to prove that he knows the representation of O_1 in the bases B and h using the protocol proposed by Okamoto [27]. In other words, he will have to prove that he knows a pair (β, α) such that $O_1 = B^\beta h^\alpha$: $\Pi_2 = \textsc{Pok}[\alpha, \beta : M_1 = g_1^\alpha \wedge N_1 = g_2^\alpha \wedge O_1 = B^\beta h^\alpha]$
- (Π_3) A proof that the voter knows the plaintext related to the ciphertext $C_3 = E_T[B^{rs^{-1}}] = (M_2, N_2, O_2)$. In particular, the voter will have to prove that he knows the representation of O_2 in the bases B and h using the protocol proposed by Okamoto [27]. In other words, he will have to prove that he knows a pair (θ, δ) such that $O_2 = B^\theta h^\delta$: $\Pi_3 = \textsc{Pok}[\delta, \theta : M_2 = g_1^\delta \wedge N_2 = g_2^\delta \wedge O_2 = B^\theta h^\delta]$
- (Π_4) A proof that the voter knows the plaintext related to the ciphertext $C_4 = E_T[g_3^x] = (M_3, N_3, O_3)$: $\Pi_4 = \textsc{Pok}[\lambda, \mu : M_3 = g_1^\lambda \wedge N_3 = g_2^\lambda \wedge O_3 = g_3^\mu h^\lambda]$
- (Π_5) A proof that the plaintext of $C_2 = E_T[B^{s^{-1}}]$ is different from 1, as explained in Section 3.2: $\Pi_5 = \textsc{Pok}[\alpha, \beta : M_1 = g_1^\alpha \wedge N_1 = g_2^\alpha \wedge O_1 = B^\beta h^\alpha \wedge \beta \neq 0 \bmod p]$
- (Π_6) A proof that the voter knows the discrete logarithm of $O = o^x$ in the basis o and that it is equal to the discrete logarithm of the plaintext of $C_4 = E_T[g_3^x]$ in the basis g_3: $\Pi_6 = \textsc{Pok}[\lambda, \mu : M_3 = g_1^\lambda \wedge N_3 = g_2^\lambda \wedge O_3 = g_3^\mu h^\lambda \wedge O = o^\mu]$

Remark: All these proofs of knowledge may be accomplished using standard techniques such as the ones mentioned in Section 3.2. As is standard practice, the challenge values for these proofs of knowledge are constructed using a call to a cryptographic hash function (the Fiat-Shamir heuristic [16]), modeled in our security analysis by a random oracle. The inputs to this cryptographic hash function for these challenges values should include $ID_{Election}$ (a random election identifier), B, $C_1 = E_T[v]$, $C_2 = E_T[B^{s^{-1}}]$, $C_3 = E_T[B^{rs^{-1}}]$, $C_4 = E_T[g_3^x]$,

$O = o^x$ and other values required for the realization of these non-interactive zero-knowledge proofs. In this way, the actual vote $C_1 = E_T[v]$ submitted by a voter will be bound to the remaining voting material. Observe that, in contrast to the scheme of JCJ that employs the plaintext equivalence test [20] to eliminate duplicates, i.e. votes posted with the same credential, our scheme uses the value o^x to perform the identification of these votes. This ensures that just one vote per credential will be processed in the tabulation phase.

Tabulation Phase. After the end of the voting period, the talliers T read all tuples $\langle E_T[v], B, E_T[B^{s^{-1}}]\ , E_T[B^{rs^{-1}}], E_T[g_3^x], o^x, \Pi\rangle$ posted on the bulletin board and process them as follows:

1. **Checking proofs:** T checks all proofs Π_i's that compose the tuples and discards tuples with incorrect proofs. In other words, T verifies that $E_T[v]$ contains a vote for a valid candidate, checks the proof of knowledge of the plaintexts with regards to $\langle E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}]\rangle$, the proof that $E_T[B^{s^{-1}}]$ does not encrypts the plaintext 1, the proof of knowledge of the plaintext related to the ciphertext $C_4 = E_T[g_3^x]$, as well as the equality of the discrete logarithm of the plaintext of $C_4 = E_T[g_3^x]$ in the basis g_3 and of o^x in the basis o. The tuples that passed the test are processed in the next step without Π and B, that is, it now contains the values $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x], o^x\rangle$;
2. **Eliminating duplicates:** In order to detect and remove tuples posted with the same credential (i.e. duplicates), T compares all o^x by means of a hash table. If a duplicate is detected, T keeps the last posted tuple (based on the order of posting on the bulletin board) and removes the other ones. T processes in the next step the values $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}],\ E_T[g_3^x]\rangle$;
3. **Mixing:** T now sends all tuples composed of $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x]\rangle$ to a verifiable mix net. The mix net outputs a permuted and re-encrypted set of tuples $\langle E_T[v]', E_T[B^{s^{-1}}]'\ , E_T[B^{rs^{-1}}]', E_T[g_3^x]'\rangle$, where $E_T[X]'$ means the re-encryption of $E_T[X]$.
4. **Checking credentials:** From the mixed tuples $\langle E_T[v]', E_T[B^{s^{-1}}]', E_T[B^{rs^{-1}}]', E_T[g_3^x]'\rangle$, the talliers T perform along with the registrars R the following steps (for each tuple) to identify the valid votes (that is the ones which *encrypt* valid credentials $\sigma = (A, r, x)$ satisfying the relation $A^{y+r} = g_1 g_3^x$):
   (a) By means of his secret key y, R cooperatively computes: $E_T[B^{s^{-1}}]'^y = E_T[B^{ys^{-1}}]'$. Now R uses the El Gamal homomorphic property to compute: $E_T[B^{ys^{-1}}]' \cdot E_T[B^{rs^{-1}}]' = E_T[B^{ys^{-1}+rs^{-1}}]'$
   (b) T now computes $C = E_T[B^{ys^{-1}+rs^{-1}} g_1^{-1} g_3^{-x}]'$ from $E_T[B^{ys^{-1}+rs^{-1}}]'$, $E_T[g_3^x]'$, and the public parameter g_1. Note that if we denote by $A = B^{s^{-1}}$ then $E_T[B^{ys^{-1}+rs^{-1}}]' = E_T[A^{y+r}]'$. Hence, $C = E_T[A^{y+r} g_1^{-1} g_3^{-x}]'$.
   (c) In order to identify a valid credential, T executes a Plaintext Equivalence Test in order to determine whether C is an encryption of the plaintext 1 or not. For this, T cooperatively selects a random number $z \in \mathbb{Z}_p$ and

computes C^z. T then decrypts C^z. If the decryption result is equal to 1, the credential is a valid one. Otherwise, the result will be a random number and this indicates an invalid credential.

5. **Tallying:** T discards all tuples with invalid credentials and cooperatively decrypts the value $E_T[v]'$ of the tuples with valid credentials.

4 Security Analysis

In this section, we define the security properties our scheme provides. Following JCJ [23], we will however explain why our protocol satisfies the standard security requirements i.e. correctness, democracy, verifiability and coercion-resistance.

Correctness. The purpose of the proofs Π_i's is to ensure that only ballots (tuples) which contain valid credentials will be counted. Indeed, when the talliers along with the registrars will perform the test described at Step 4 of the Tabulation Phase to determine whether the ballot contains a valid credential or not, they will compute the following ciphertext which is, using the above notation (see the **Voting Phase**), equal to $E_T[B^{y\beta}B^{\theta}g_1^{-1}g_3^{-\mu}]$. In other words this is an encryption of $B^{y\beta+\theta}g_1^{-1}g_3^{-\mu}$. If this is an encryption of 1, this means that $B^{y\beta+\theta}g_1^{-1}g_3^{-\mu} = 1$. Remember that the voter has also to prove that $\beta \neq 0 \pmod{p}$ in the voting phase (using the proof of knowledge owing to Camenisch and Shoup [10]). Let us denote by $A = B^{\beta}$, $r = \theta/\beta$ and $x = \mu$. So $B^{y\beta+\theta}g_1^{-1}g_3^{-\mu} = 1$ can be rewritten as follows: $A^{y+r}g_1^{-1}g_3^{-x} = 1$ which is equivalent to $A^{y+r} = g_1g_3^x$. In other words, if all the zero-knowledge proofs are valid, this means that the voter knows a tuple $(A,\ r,\ x)$ such that $A^{y+r} = g_1g_3^x$. Therefore he knows a valid credential. The proof that $\beta \neq 0 \pmod{p}$ is crucial. Without this proof, an adversary could generate a ballot that will pass the test, without knowing a valid credential. Thanks to this proof, only ballots that encrypt a valid credential will pass this test.

The method employed in this verification ensures therefore that a valid vote cannot be identified as invalid or vice versa. In addition, no one can produce valid credentials as this would involve breaking the q-SDH assumption. Therefore, only the votes from eligible voters will appear in the final count.

Democracy. By removing duplicates in step 2, we ensure that only one vote per credential (valid or fake) is processed in the remaining steps. This is performed by comparing all values o^x and then by verifying two or more values o^x match. If this takes place, only the last posted vote is considered in the next step; the others are discard. The fact that only one valid credential is processed in the next steps ensures that one vote per eligible voter will be counted.

Universal Verifiability. The bulletin board and the NIZKPs allow anyone to verify the tuples and the work of the talliers. Each posted tuple contains a set of NIZKPs. These proofs allow anyone to verify that the tuple is well-formed. Anyone can perform this and complain to the authorities in case where tuples with invalid proofs are sent to the next step. In the same way, after processing

the tuples in each step of the tallying phase, the authorities publish proofs on the bulletin board. In step 2, for example, as all the tuples are published, anyone can perform the comparisons. In addition, the mix net provide NIZKPs after processing the tuples in step 3; the authorities publish NIZKPs after verifying the credentials in step 4*c*; especially, the authorities prove that they have used the correct share after powering the encrypted credential to a random number.

Coercion-resistance. Our voting protocol satisfies the coercion-resistance requirement as defined in [23].

Theorem 1. *The proposed voting protocol satisfies the coercion resistance requirement, in the random oracle model, under the q-SDH and SDDHI assumptions.*

Owing to space limitations, the proof of this theorem is omitted from this extended abstract and will appear in the full version of the paper.

5 Conclusion

In this paper, we have introduced a new coercion-resistant scheme. Our solution (which has a linear work factor instead of a quadratic work factor in previous solutions) is practical and secure. It employs anonymous credentials that have a similar structure than the membership certificates in the group signature scheme of Boneh, Boyen, and Shacham [5]. These credentials have their security based on the *q-Strong Diffie-Hellman* and on the *Strong Decisional Diffie-Hellman Inversion* assumptions.

Differently from some previous schemes, the new credentials can be used in more than one election. This way, a voter does not need to issue a new credential each time a new election takes place. We have also shown that a recent coercion-resistant voting protocol is insecure.

References

1. Araújo, R.: On Remote and Voter-Verifiable Voting. PhD thesis, Technische Universität Darmstadt, Darmstadt, Germany (September 2008)
2. Araújo, R., Foulle, S., Traoré, J.: A practical and secure coercion-resistant scheme for remote elections. In: Chaum, D., Kutylowski, M., Rivest, R.L., Ryan, P.Y.A. (eds.) Frontiers of Electronic Voting, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, vol. 07311, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany (2008)
3. Boneh, D.: The decision diffie-hellman problem. In: Buhler, J. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
4. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
6. Brickell, E.F. (ed.): CRYPTO 1992. LNCS, vol. 740. Springer, Heidelberg (1993)

7. 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)
8. Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., Meyerovich, M.: How to win the clone wars: efficient periodic n-times anonymous authentication. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM Conference on Computer and Communications Security, pp. 201–210. ACM, New York (2006)
9. Camenisch, J., Kiayias, A., Yung, M.: On the portability of generalized schnorr proofs. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 425–442. Springer, Heidelberg (2009)
10. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
11. Canetti, R., Gennaro, R.: Incoercible multiparty computation (extended abstract). In: FOCS, pp. 504–513 (1996)
12. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM 24(2), 84–88 (1981)
13. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell [6], pp. 89–105
14. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: IEEE Symposium on Security and Privacy, pp. 354–368. IEEE Computer Society, Los Alamitos (2008)
15. Clarkson, M.R., Myers, A.C.: Coercion-resistant remote voting using decryption mixes. In: Workshop on Frontiers in Electronic Elections (2005)
16. 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 (1986)
17. Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Transferable constant-size fair e-cash. Cryptology ePrint Archive, Report 2009/146 (2009), `http://eprint.iacr.org/`
18. Furukawa, J., Sako, K.: An efficient publicly verifiable mix-net for long inputs. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 111–125. Springer, Heidelberg (2006)
19. 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 (1984)
20. 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)
21. Jefferson, D., Rubin, A., Simons, B., Wagner, D.: A security analysis of the secure electronic registration and voting experiment (2004)
22. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. Cryptology ePrint Archive, Report 2002/165 (2002), `http://eprint.iacr.org/`
23. 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)
24. Kuesters, R., Truderung, T., Vogt, A.: A game-based definition of coercion-resistance and its applications. Cryptology ePrint Archive, Report 2009/582 (2009), `http://eprint.iacr.org/`

25. Moran, T., Naor, M.: Receipt-free universally-verifiable voting with everlasting privacy. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 373–392. Springer, Heidelberg (2006)
26. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: ACM Conference on Computer and Communications Security, pp. 116–125 (2001)
27. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell (ed.) [6], pp. 31–53 (1992)
28. Schnorr, C.-P.: Efficient signature generation by smart cards. J. Cryptology 4(3), 161–174 (1991)
29. Schweisgut, J.: Coercion-resistant electronic elections with observer. In: Krimmer, R. (ed.) Electronic Voting. LNI, vol. 86, pp. 171–177. GI (2006)
30. Smith, W.: New cryptographic election protocol with best-known theoretical properties. In: Workshop on Frontiers in Electronic Elections (2005)
31. Unruh, D., Müller-Quade, J.: Universally composable incoercibility. Cryptology ePrint Archive, Report 2009/520 (2009), `http://eprint.iacr.org/`
32. 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)