A Practical Coercion Resistant Voting Scheme Revisited

Roberto Araújo[1] and Jacques Traoré[2]

[1] Universidade Federal do Pará, Faculdade de Computação, Rua Augusto Corrêa 01, 66075-110, Belém/PA, Brazil
rsa@ufpa.br

[2] Orange Labs, 42 rue des Coutures, BP 6243, 14066 Caen Cedex, France
jacques.traore@orange.com

Abstract. The scheme of ABRTY (Araújo et al., CANS 2010) is one of the most promising solutions for internet voting nowadays. It fights realistic coercive attacks and can be applied in large scale voting scenarios as it has linear time complexity. However, this scheme has two intrinsic drawbacks. As it does not allow revocation of credentials of ineligible voters, voters need to obtain fresh credentials before each new election. Also, authorities could generate valid but illegitimate credentials unnoticed. In this work, we present solutions for these drawbacks and show a modified version of ABRTY's scheme. In addition, we describe a weakness of a receipt-free voting scheme proposed by Acquisti in 2004.

1 Introduction

A number of researchers consider internet voting as impracticable. The reason is the risks inherent to its environment. Malwares, for instance, may infect voters' computers to break the secrecy of the vote. Hackers, in turn, could compromise voting systems around the world. In addition, due to its uncontrolled environment, internet voting is susceptible to coercion and vote-selling.

In order to deal with these problems, Juels, Catalano, and Jakobsson (JCJ) [17] introduced the first coercion-resistant scheme that considers a powerful adversary. This adversary cannot succeed in issuing receipts (receipt-freeness), threatening voters to abstain from voting, revealing private data, or even casting random votes.

The idea behind JCJ's proposal is that a voter is able to deceive adversaries about her true vote intention. That is, the voter receives a valid credential (e.g. an alphanumeric string) in a secure manner. When she wants to cast her vote, she uses this credential. A voter under coercion, however, can make a fake credential and give it to the coercer. Later on, she can vote again using her valid credential. The coercer has no way to distinguish between the credentials used. Unfortunately, the earliest implementation of this idea is inefficient for large scale voting scenarios. This idea, though, inspired efficient solutions such as Araújo et al. [3], Spycher et al. [25], and Essex et al. [13].

J. Heather, S. Schneider, and V. Teague (Eds.): VoteID 2013, LNCS 7985, pp. 193–209, 2013.

One of the requisites of coercion-resistant voting schemes as JCJ is that a voter has to receive her credential in a trustworthy way. As this takes into account a untappable channel between the registrars and the voter, this process is supposed to be performed in a registration place. At this place, she obtains her credential after proving to the registrars that she is eligible for voting in an election.

A desirable property of credentials is to employ them in more than one election. In other words, once the voter obtained her credential, she can use it in several upcoming elections. This reduces costs and affords more convenience as voters do not need to revisit the registration place. However, it is necessary to prevent a voter from participating in an election for which she is not eligible for. To this end, authorities must be able to revoke credentials of these voters.

In contrast, if credentials cannot be used in a second election or more, voters need to visit the registration place again before each new election. This would be impractical for large scale voting scenarios.

Based on JCJ's proposal, ABRTY [3] introduced one of the most promising solutions for internet voting nowadays. The scheme is efficient for large scale scenarios as it has linear time complexity and was shown secure. ABRTY's scheme, though, has two intrinsic drawbacks that can discourage its use.

In ABRTY's solution, authorities have no way to revoke credentials. That is, they cannot disallow a voter in a specific election without performing this for all voters. As a consequence, all voters need to revisit the registration place to obtain fresh credentials.

Another drawback of ABRTY's scheme is that a majority of malicious registrars could generate valid (but illegitimate) credentials without being noticed. This means that votes cast using these credentials could appear in the final tally. Schläpfer et al. [21] and Spycher et al. [25] pointed out this problem.

These are important drawbacks of ABRTY's solution, but fortunately they can be fixed. In this work, we introduce techniques to fix these drawbacks and present a modified version of ABRTY's scheme accordingly. We also show that Acquisti's protocol [1] is not coercion-resistant and that its credentials can only be used once.

This text is organized as follows: the next section presents an overview of ABRTY's scheme and its drawbacks. Section 3 shows the improvements to ABRTY's solution and introduces an improved variant scheme. Finally, we present our conclusions in Section 4. In Appendices, we briefly describe the scheme of Acquisti and show its weaknesses.

Related Work on Coercion-Resistance

Juels, Catalano, and Jakobsson (JCJ) presented the first coercion-resistant scheme in 2005 at WPES [17]. Following JCJ's work, several coercion-resistant schemes were proposed. Acquisti [1] introduced a scheme in which the vote and the credential are combined, and which strongly relies on homomorphisms. Meng [18] proposed a solution similar to Acquisti's protocol. Clarkson et al. [11] presented

a variant of Prêt-à-Voter [7] scheme suitable for internet voting and based on decryption mix nets.

Schweisgut [23] and more recently Clarkson et al. [10] proposed schemes which mitigate the inefficiency problem of the JCJ's solution. The former scheme relies on decryption mix nets and a tamper-resistant hardware, whereas the latter is a modified version of JCJ's proposal. ABRTY [3] showed that the scheme of Schweisgut is not coercion-resistant.

Smith [24] presented an efficient scheme with linear work factor. Weber et al. [28], 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, both solutions are not coercion-resistant as shown by Araújo et al. [2]. The problem of Smith's scheme was also noted independently by Clarkson et al. [10].

Araújo, Foulle, and Traoré [2] presented the first practical and secure coercion-resistant scheme. Their proposal, different from the previous ones, employs credentials with a special structure that allows the scheme to achieve a linear work factor. Based on [2], ABRTY [3] introduced a different scheme that uses new anonymous credentials derived from the group signature scheme of Boneh et al. [4] and formally proved that their scheme is coercion-resistant.

From Araújo et al.'s proposals [2,3], other promising efficient schemes appeared. Spycher et al. [25] introduced a protocol where voters indicate their credentials on an electoral roll. Spycher et al. [26] improved [25] later on. Clark et al. [9] introduced a proposal that uses special passwords as credentials. Schläpfer et al. [21] showed a scheme that links the votes to the voter roll.

However, the complexity of the schemes of Spycher et al. [25], Clark et al. [9], and Schläpfer et al. [21] depends on a security parameter β to balance efficiency against coercion-resistance: the complexity is in $O(\beta N)$ for Clark et al. [9] and Schläpfer et al. [21] and in $O(N + \beta n)$ for Spycher et al. [25], where N is the number of submitted ballots and n is the number of registered voters.

More recently, Essex at al. [13] introduced a promising scheme that authorizes votes during the voting phase. Unfortunately, as pointed out by the authors, this scheme is not efficient for real world elections as it "requires a registration process that is quadratic in the number of eligible voters".

As the schemes of Spycher et al. [25], Clark et al. [9], and Schläpfer et al. [21] depend on a security parameter and the solution of Essex at al. [13] has an inefficient registration phase, only the schemes of Araújo et al. [2,3] are truly linear in N (see Schläpfer et al. [21] for a detailed performance comparison analysis of these schemes). In addition, the security of [2,25,9,21,13] are only conjectured.

2 The ABRTY's Scheme and Its Drawbacks

In this section, we briefly introduce ABRTY's scheme [3]. In addition, we recall a drawback presented by Spycher et al. [25] and introduce another one.

2.1 An Overview of ABRTY's Proposal

The scheme of ABRTY follows the principle introduced by JCJ. The voter receives a credential in a secure manner. She uses it to cast her valid vote and may generate fake credentials to deceive adversaries. However, the credential employed in ABRTY's solution, which is based on Boneh et al.'s group signature scheme (BBS for short) [4], has a mathematical structure that ensures its security.

Cryptographic Building Blocks. ABRTY's scheme requires a set of cryptographic building blocks. It relies on a threshold El Gamal cryptosystem and a universally verifiable mix net. In addition, the scheme requires several zero-knowledge proof protocols. It prevents adversaries from using El Gamal malleability by means of Schnorr signatures [22]. It uses a protocol to prove that a ciphertext contains a vote for a valid candidate such as the proposal of Hirt and Sako [14]. ABRTY's solution also uses a protocol for proving the equality of discrete logarithms relative to different bases owing to Chaum and Pedersen [6], a protocol for proving knowledge of a representation as Okamoto [19] and a plaintext equivalence test [15]. Also, it requires a Web Bulletin Board (WBB). The security of its credentials depends on the q-Strong Diffie-Hellman (q-SDH) [4] and the Strong Decisional Diffie-Hellman Inversion (SDDHI) assumptions [3].

The Scheme. It considers a set of voters, a set of registrars that make valid credentials for the voters and help the talliers in the tallying phase, and a set of talliers that compute the final tally. Let $E_X(Y)$ be the encryption of Y with the public key X. In the setup phase, a set of authorities generate the key material and publish the corresponding public parameters on a WBB. In particular, the authorities define a cyclic group $\mathbb{G}$ with prime order p and four generators $g_1, g_2, g_3, o \in \mathbb{G}$. The Decision Diffie-Hellman problem (DDH) must be hard in $\mathbb{G}$. The talliers cooperate to generate a key pair by means of the threshold El Gamal cryptosystem, i.e. the public key T and its corresponding private key $\widehat{T}$. The registrars also cooperate to generate a key pair, namely the public key $R = g_3^y$ and its corresponding private key $\widehat{R} = y$.

Registration Phase. After proving to the registrars that she is eligible, the voter receives her credential securely, through a untappable channel. In order to compute a credential, the registrars select two random numbers $r, x \in \mathbb{Z}_p$ and compute: $A = (g_1 g_3^x)^{\frac{1}{y+r}}$, where g_1, g_3 are public generators and y is the secret key of the registrars. The credential σ is composed of three parts, namely $\sigma = (A, r, x)$. The voter must keep the value x in secrecy. The parts (A, r) can be stored in a device or be sent by email to the voter without compromising the security of the scheme.

Voting Phase. The voter selects a random number $s \in \mathbb{Z}_p$, uses her credential $\sigma = (A, r, x)$ to compute $B = A^s$ and makes a tuple containing: her encrypted vote, her encrypted credential and a set of non-interactive zero-knowledge proofs (NIZKPs). In other words, she makes the tuple $\langle E_T[v], B,$

$E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x], o^x, P\rangle$, where v is her vote intention, $B = A^s$, g_3 and o are public generators and P is a set of NIZKPs. This set is composed of a proof that $E_T[v]$ contains a vote for a valid candidate, proofs that the voter knows the plaintext underlying the ciphertexts, a proof that the plaintext of $E_T[B^{rs^{-1}}]$ is different from 1 as well as a proof that the discrete logarithm of o^x in the basis o is equal to the discrete logarithm of the plaintext of $E_T[g_3^x]$ in the basis g_3. The voter then uses an anonymous channel to send her tuple to the WBB and thereby casting her vote.

Tallying Phase. The tallying takes place after the voting phase. It comprises 5 steps. The talliers first read all tuples posted on the WBB and verify all proofs on each tuple. They discard tuples with invalid proofs and then process the tuples that passed the tests. A tuple now has the following format: $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x], o^x\rangle$. The talliers then eliminate tuples posted using the same credential (i.e duplicates). For this, they compare all o^x by means of a hash table. If the talliers detect a duplicate, they use the time of posting on the WBB to verify the last posted tuple. The talliers keep the last posted tuple that now is composed of: $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x]\rangle$. After that, they send all tuples $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x]\rangle$ to a verifiable mix net. The mix net output a set of permuted and re-encrypted tuples of the form: $\langle E_T[v]', E_T[B^{s^{-1}}]', E_T[B^{rs^{-1}}]', E_T[g_3^x]'\rangle$, where $E_T[\cdot]'$ is the reencryption of $E_T[\cdot]$.

The talliers now check the valid credentials. For this, they perform the following steps: (1) they instruct the registrars to cooperatively compute $E_T[B^{s^{-1}}]'^y = E_T[B^{ys^{-1}}]'$ and $E_T[B^{ys^{-1}}]' \cdot E_T[B^{rs^{-1}}]' = E_T[B^{ys^{-1}+rs^{-1}}]'$ by using the secret key y; (2) the talliers compute $C = E_T[B^{ys^{-1}+rs^{-1}} g_1^{-1} g_3^{-x}]'$ from $E_T[B^{ys^{-1}+rs^{-1}}]'$, $E_T[g_3^x]'$, and the public parameter g_1. (3) they now cooperatively select a random number $z \in \mathbb{Z}_p$, compute C^z and decrypt C^z. If they obtain 1 after decrypting C^z, the credential is valid; otherwise, the credential is invalid. In order to finish the tallying phase, the talliers discard all tuples with invalid credentials, cooperatively decrypt $E_T[v]'$ of the tuples with valid credentials, and publish the voting results on the WBB.

2.2 The Drawbacks

The scheme of ABRTY was shown secure in the random oracle model, under the q-Strong Diffie-Hellman and Strong Decisional Diffie-Hellman Inversion assumptions. It can be used in large scale voting scenarios as it has a linear time complexity. Unfortunately, this scheme has two intrinsic drawbacks.

As Spycher et al. [25] presented, "a majority of colluding registrars could compute valid (but illegitimate) credentials unnoticed". In other words, after proving that she is eligible, each voter receives a valid credential from the registrars. Although an adversary cannot forge a credential after the registration because this requires breaking the q-SDH assumption, a majority of registrars could act as adversaries during this phase. These malicious registrars could make valid (but illegitimate) credentials as the scheme does not verify whether they generated a

credential to an eligible voter (i.e. a legitimate credential) or not. Thus, a valid (but illegitimate) credential would pass the tests performed in the tallying phase such that the corresponding vote would be counted.

Besides the drawback presented by Spycher et al., the scheme has another one: it does not allow revocation of credentials. In any new election, the number of eligible voters may change. Some voters may have their right to vote revoked after participating 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, authorities must be able to revoke credentials when necessary.

The credentials employed in ABRTY's scheme may be used in several elections as long as the same key y is employed. However, in principle a credential cannot be revoked. Because only the voters know their credentials, the authorities are not able to revoke a credential. In addition, even if the authorities store all credentials and put them in a black list, this would be of no help. There is no way in the tallying phase to decide whether a revoked credential has been used or not. Indeed the credentials are published on the WBB in an encrypted form. Clark et al. [9] and Essex et al. [13] pointed out a similar drawback.

3 Improving ABRTY's Scheme

As presented above, ABRTY's scheme does not allow authorities to revoke credentials. In addition, a majority of colluding registrars could issue valid but illegitimate credentials. Aiming at eliminating these drawbacks, we present here new mechanisms that improve ABRTY's solution. Also, we introduce a version of ABRTY's scheme that employs our mechanisms.

3.1 Revoking Credentials

Although the design of ABRTY's scheme makes revocation difficult, it has some properties that help accomplish this. Recall from ABRTY's scheme that, upon registering, a voter receives a credential $\sigma = (A, r, x)$. The element x must be transmitted via an untappable channel. Conversely, the elements $\langle A, r\rangle$ can be sent by post or even by email. This does not compromise the credential's security as long as the SDDHI assumption holds.

Based on this, it is possible to use a similar method as the technique employed by Boneh et al. [4] to revoke membership certificates in their group signature scheme. That is, in order to revoke credentials and perform new elections, the authorities could execute the following steps:

Besides generating and issuing a credential to a voter, the registrars store the *public part* (A, r) of the credential in a list L_C. Suppose we wish to revoke the credential of a voter V^*. To perform this, the registrars first retrieves from L_C the public part of the credential of V^* (i.e. (A^*, r^*)) and then update their public key. The new public key is $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R})$ where $\widetilde{g}_1 = g_1^{1/(y+r^*)}, \widetilde{g}_3 = g_3^{1/(y+r^*)}$ and $\widetilde{R} = \widetilde{g}_1^{y}$. The secret key y remains unchanged. The registrars then publish the values $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R}, r^*)$ in a revocation list RL.

Let us show now how unrevoked voters update their credentials. Consider an unrevoked voter whose credential is (A, r, x). Given $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R}, r^*)$ obtained from RL, the voter computes $\widetilde{A} = \widetilde{g}_1^{\frac{1}{r-r^*}} \widetilde{g}_3^{\frac{x}{r-r^*}} A^{\frac{-1}{r-r^*}}$ and sets her new credential to be $(\widetilde{A}, r, x)$. Notice that since the values r and r^* are randomly chosen by the registrars, then with high probability we have $r \neq r^* \bmod p$.

One can verify that $(\widetilde{A}, r, x)$ is a valid credential for the new public key $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R})$ by computing:

$$\begin{aligned}
\widetilde{A}^{y+r} &= \widetilde{g}_1^{\frac{y+r}{r-r^*}} \widetilde{g}_3^{\frac{(y+r)x}{r-r^*}} A^{\frac{-(y+r)}{r-r^*}} \\
&= \widetilde{g}_1^{\frac{y+r}{r-r^*}} \widetilde{g}_3^{\frac{(y+r)x}{r-r^*}} g_1^{\frac{-1}{r-r^*}} g_3^{\frac{-x}{r-r^*}} \\
&= \widetilde{g}_1^{\frac{y+r}{r-r^*}} \widetilde{g}_3^{\frac{(y+r)x}{r-r^*}} \widetilde{g}_1^{\frac{-(y+r^*)}{r-r^*}} \widetilde{g}_3^{\frac{-(y+r^*)x}{r-r^*}} \\
&= \widetilde{g}_1 \widetilde{g}_3^x
\end{aligned}$$

This process can be repeated several times if there are more than one credential to revoke. Using similar arguments as the ones given in [4], one can prove that under the q-SDH assumption the revoked voter V^* cannot construct a valid credential for the new public key $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R})$. However, if the revoked voter V^* was under coercion in a previous election, then the coercer can now check using the new public key $(\widetilde{g}_1, \widetilde{g}_3, \widetilde{R})$ if V^* previously revealed a fake credential or a valid one. Indeed, if V^* gave him (A^*, r^*, x'), he just has to test whether $A^* \stackrel{?}{=} \widetilde{g}_1 \widetilde{g}_3^{x'} = g_1^{1/(y+r^*)} g_3^{x'(1/(y+r^*))}$.

We therefore propose another method in order to avoid this problem. It allows credentials of eligible voters to be used in more than one election as follows:

- In the *setup phase*, besides generating their key pair (namely $R = g_3^y$ and $\widehat{R} = y$) as usual, the registrars cooperatively make a new key pair: $\langle \mathcal{R}, \widehat{\mathcal{R}} \rangle$. $\mathcal{R}$ is the public key and $\widehat{\mathcal{R}}$ is its corresponding shared private key. This key pair will be used for the purpose of revocation later on.
- During the *registration phase*, the registrars cooperatively generate a credential $\sigma = (A, r, x)$ for each voter. After that, the registrars use the public key $\mathcal{R}$ to cooperatively compute the encryption of $g_1 g_3^x$ (namely $E_{\mathcal{R}}[g_1 g_3^x]$) and store this ciphertext in a list L_1.
- For each new election, instead of using the same shared private key $\widehat{R} = y$, the registrars generate a new shared private key y' (i.e. now $R = g_3^{y'}$ and $\widehat{R} = y'$) and furnish the voters with new values $\langle A' = (g_1 g_3^x)^{\frac{1}{y'+r'}}, r' \rangle$. They compute this from the values $E_{\mathcal{R}}[g_1 g_3^x]$ stored in L_1 and from a randomly chosen value r'. That is, $A' = (g_1 g_3^x)^{\frac{1}{y'+r'}}$ is cooperatively computed by raising $E_{\mathcal{R}}[g_1 g_3^x]$ to the power $\frac{1}{y'+r'}$ using the technique of Wang et al. [27]. After that, $E_{\mathcal{R}}[(g_1 g_3^x)^{\frac{1}{y'+r'}}]$ is cooperatively decrypted to retrieve A'.

– The new credential is the tuple $\langle A', r', x\rangle$ where x is the secret value obtained by the voter during her first registration. The others elements of the credential (i.e. A' and r') could be sent by mail to the voter or even published on a dedicated website.

This novel method makes possible the revocation of credentials in ABRTY's scheme. For each new election, the authorities need a new shared private key y and the eligible voters receive fresh values $\langle A', r'\rangle$. Voters who belong to a revocation list or are not allowed to vote will not receive an updated credential. The voter does not have to revisit the registration place and register again in order to obtain her new credential. She can download the new parts of her credential $\langle A', r'\rangle$ from WBB, for instance. To complete her credential, the voter employs the same x of the credential that she received when she registered for the first time. Thus, the voter does not need to replace the whole credential (i.e. $\sigma = (A, r, x)$) before each new election.

Note that, if a malicious voter has a revoked credential $\langle A_m, r_m, x_m\rangle$ and obtains the new values $\langle A', r'\rangle$ of another voter, she may use the credential $\langle A', r', x_m\rangle$ to vote. However, the vote tuple computed with this credential will not pass the tests in the tallying phase. When the authorities use $E_{\mathcal{R}}[g_1 g_3^x]$ to generate the new parts $\langle A' = (g_1 g_3^x)^{\frac{1}{y'+r'}}, r'\rangle$ of the credential to a voter, they employ the same exponent x that the voter received in the registration phase. As the malicious voter does not have the correct exponent x, her vote tuple will be removed from the tally during the verification process. That is, to verify the valid credentials, the authorities use the values $E_T[g_3^{x_m}], E_T[B^{s^{-1}}], E_T[B^{r's^{-1}}]$ from the malicious voter's tuple and compute $C = E_T[B^{y's^{-1}+r's^{-1}}\ g_1^{-1} g_3^{-x_m}]'$ next, where $B = (A')^s = ((g_1 g_3^x)^{\frac{1}{y'+r'}})^s$ for a random s. By performing this, the authorities obtain the ciphertext C that encrypts $g_3^{x-x_m}$ instead of the value 1. This will make the tuple invalid after the authorities apply the plaintext equivalence test.

3.2 Defeating a Majority of Colluding Registrars

ABRTY's scheme allows the computation of valid (but illegitimate) credentials by a majority of malicious registrars. These fraudulent credentials could be used for ballot stuffing. Although this drawback reduces the security of the scheme, fortunately it can be fixed. We present now a method that identifies any valid (but illegitimate) credential and that allows removing these credentials from the tallying. The method was inspired by the works of Smith [24] and Weber et al. [28].

Comparison and Computation of Fingerprints. Before introducing our solution, we briefly recall the method of comparison and computation of fingerprints owing to Weber et al. This technique is used to blind and compare plaintexts inside ciphertexts without leaking any information about the true plaintexts. The method works as follows: in order to blind a plaintext of an El

Gamal ciphertext, a set of n authorities jointly generate a secret shared value z. These authorities then apply their shares of z to each ciphertext to blind the corresponding plaintext. The authorities now decrypt all ciphertexts to obtain the blinded plaintexts (i.e. fingerprints). For example, from a ciphertext $C = E_T[m]$ of a message m, the fingerprint will be equal in this case to m^z. To compare the fingerprints, the authorities can use a hashtable algorithm.

Taking into account Weber et al. technique as a building block, we introduce our solution as follows:

- In the *registration phase*, after computing a credential $\sigma = (A, r, x)$ to the voter, the registrars cooperatively compute: $E_T[A]$, where T is the public key of the talliers. The registrars may issue a designated verifier proof [16] to prove that $E_T[A]$ encrypts the same A of the voter credential.
- Let $\mathrm{ID}_{\mathrm{voter}}$ be a voter unique identification number (i.e. a number associated to the voter's name in the electoral roll) and L_2 be a public list of legitimate credentials. For each voter, the registrars store in L_2 the pair $\langle E_T[A], \mathrm{ID}_{\mathrm{voter}}\rangle$.
- In order to check that only legitimate credentials have been used in the *tallying phase*, the talliers proceed as follows:
  - After eliminating duplicates and invalid ballots (i.e. those with invalid proofs or invalid credentials), the talliers obtain the list LV containing the remaining ballots.
  - The talliers send the ciphertexts $E_T[A]$ of L_2 to a verifiable mix net and obtain L_2';
  - By means of Weber et al.'s technique, the talliers raise all the ciphertexts in L_2' and LV (only the values $E_T[A]$) to a random secret exponent and decrypt the resulting ciphertexts. Let L_2'' and LV' denote the new lists;
  - The talliers use a hashtable to verify whether every element in LV' belongs to L_2''. If there is an element in LV' that does not match with an element in L_2'', then this means that either someone has broken the Boneh et al. [4] group signature scheme or that the registrars produced valid but illegitimate credentials. In any case, the talliers remove ballots corresponding to illegitimate credentials from the list L_2''.

3.3 The Improved Variant of ABRTY's Scheme

Based on the mechanisms introduced above, we describe now an improved version of ABRTY's scheme. This version eliminates the drawbacks of the original proposal while keeping the same security properties of it.

The new version employs the same cryptographic primitives as the original one (see Section 2.1). In addition, the new scheme requires the global blind comparison mechanism due to Weber et al. [28] and the method of Wang et al. [27]. It has four phases: setup, registration, voting, and tallying.

However, there are two ways to perform the setup and the registration phases. For a first election, the authorities execute these phases for the first time. All voters need to visit the registration place once to obtain their credentials.

For a second election (or more), as there are previous election parameters and registered voters, the authorities execute different setup and registration phases. Voters that registered before do not need to visit the registration place again. Authorities, in turn, can revoke credentials.

Therefore, there are two states for the scheme. It can be either used in a first election or in several elections (i.e. a second election or more). The scheme is described below.

For a *first election*, the scheme requires the following setup and registration phases:

Setup phase for a first election. If the scheme is used for the first time, the voters do not have any credential and need to obtain them to vote. In this case, the authorities establish first a cyclic group $\mathbb{G}$ (the DDH must be hard in $\mathbb{G}$) with prime order p and five generators $g_1, g_2, g_3, g_4, o \in \mathbb{G}$. The talliers now cooperate to compute the El Gamal key pair for which the public key is denoted by T and the private key is denoted by $\widehat{T}$. The registrars cooperate to compute two pair of keys: a public key $R = g_3^y$ and its secret key $\widehat{R} = y$, and an El Gamal public key $\mathcal{R}$ and its secret key $\widehat{\mathcal{R}}$. The authorities publish all public material on a WBB.

Registration phase for a first election. After authenticating a voter that has an identification number $\mathrm{ID}_{\mathrm{voter}}$, the registrars select two random numbers $r, x \in \mathbb{Z}_p$ and cooperatively computes $A = (g_1 g_3^x)^{\frac{1}{y+r}}$. Then, they issue the credential $\sigma = (A, r, x)$ to the voter via an untappable channel [1]. The registrars now cooperate to encrypt $g_1 g_3^x$ with their public key $\mathcal{R}$ and store the pair $\langle E_{\mathcal{R}}[g_1 g_3^x], \mathrm{ID}_{\mathrm{voter}} \rangle$ in a list L_1 on the WBB. In addition, they encrypt A with the public key T and store the pair $\langle E_T[A], \mathrm{ID}_{\mathrm{voter}} \rangle$ in a list L_2 on the WBB.

After receiving her credential, a voter can participate in several elections. However, as the voter may not be allowed to vote in some elections, the authorities have to update the credentials of the eligible voters before each new election. It is not necessary for voters to revisit the registration place. Thus, if the scheme is used in a second election (or more), it requires slightly different setup and registration phases. These phases are presented next:

Setup phase for several elections. If the scheme is used a second time or more, all voters (or some of them) already have their credentials. In this case, the authorities (talliers and registrars) can either keep all credentials or revoke some of them. If the authorities do not need to revoke credentials, they can keep all but one parameter used in the last election. That is, they replace the generator $o \in \mathbb{G}$ used in the last election by a new one: $o' \in \mathbb{G}$. The authorities publish the new generator $o' \in \mathbb{G}$ as well as the other parameters

[1] Although we do not consider the issue of "panic passwords" [8] in this paper, we would like to emphasize that the technique introduced by Clark et al. [9] to generate such passwords also applies to our scheme.

of the last election, namely the cyclic group $\mathbb{G}$ with prime order p, the other three generators $g_1, g_2, g_3 \in \mathbb{G}$, and the key pairs $\langle T, \widehat{T}\rangle$, $\langle R, \widehat{R}\rangle$ and $\langle \mathcal{R}, \widehat{\mathcal{R}}\rangle$. Conversely, if the authorities need to revoke credentials, they replace the generator $o \in \mathbb{G}$ as before and execute an extra step. The registrars replace their key pair $\langle R = g_3^y, R = y\rangle$ used in the last election by a new one. For this, they cooperatively generate the new key pair: $\langle R = g_3^{y'}, \widehat{R} = y'\rangle$, where $\widehat{R} = y'$ is a secret key shared among the registrars. This new key pair is used to update credentials of voters allowed to vote and to generate credentials for new voters in the current election. All public material is published on the WBB.

Updating the credentials of eligible voters. If some voters are not allowed to vote in the new election, the authorities need to update the credentials of the eligible voters. In order to perform this, the registrars read from L_1 all tuples $\langle E_{\mathcal{R}}(g_1 g_3^x), \mathrm{ID}_{\mathrm{voter}}\rangle$. They created this list in the first election (see the setup phase of the first election above). By inspecting the values $\mathrm{ID}_{\mathrm{voter}}$, the registrars identify tuples of voters that can vote in the new election. For each of these tuples, the registrars cooperatively select a random $r' \in \mathbb{Z}_p$, compute $\frac{1}{y'+r'}$ using their new secret key $\widehat{R} = y'$ and power $E_{\mathcal{R}}(g_1 g_3^x)$ to $\frac{1}{y'+r'}$ by means of the method of Wang et al. [27]. Then, they cooperatively decrypt $E_{\mathcal{R}}[(g_1 g_3^x)^{\frac{1}{y'+r'}}]$ to obtain $A' = (g_1 g_3^x)^{\frac{1}{y'+r'}}$. Next, they publish $\langle \mathrm{ID}_{\mathrm{voter}}, A', r'\rangle$ on the WBB. The voters allowed to vote can now visit the WBB to identify their IDs and read their new pair $\langle A', r'\rangle$. Because each eligible voter employs the same secret x generated for the first election, the new credential is: $\langle A', r', x\rangle$. Finally, for each voter $\langle \mathrm{ID}_{\mathrm{voter}}\rangle$ allowed to vote, the registrars replace $E_T[A]$ by $E_T[A']$ in L_2.

Observe that a voter without a credential can register to vote in any election as long as she visits the registration place to obtain one. In order to issue this credential, the registrars employ their secret key $\widehat{R} = y'$ to generate the credential $\sigma = (A, r, x)$. In addition, they update the lists $L1$ and $L2$ with the values $\langle E_{\mathcal{R}}[g_1 g_3^x], \mathrm{ID}_{\mathrm{voter}}\rangle$ and $\langle E_T[A], \mathrm{ID}_{\mathrm{voter}}\rangle$, respectively.

From the setup and registration phases presented above, the voting and tallying phases are described as follows:

Voting Phase. The voter performs the same steps as in the original scheme (see Section 2). That is, she makes the tuple $\langle E_T[v], B, E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x], o^x, P\rangle$ and sends it through an anonymous channel to the WBB. Note that if the voter can vote in several elections, she uses her updated credential $\langle A', r', x\rangle$ and the new generator o'.

Tallying Phase. The talliers first read all tuples from the WBB, verify all proofs and discard tuples with invalid proofs. The remaining tuples are now

composed of $\langle E_T[v], E_T[B^{s^{-1}}], E_T[B^{rs^{-1}}], E_T[g_3^x], o^x \rangle$. After that, they eliminate tuples posted using the same credential (i.e. duplicates). In other words, the talliers use a hash table to compare all values o^x. If a duplicate is detected, they use the time of posting on the WBB to identify the last posted tuple and keep only this last posted tuple. Then, the talliers exclude the values o^x of the remaining tuples and send these new tuples to a mix net. After permuting and reencrypting these tuples, the mix net outputs tuples of the form $\langle E_T[v]', E_T[B^{s^{-1}}]' , E_T[B^{rs^{-1}}]', E_T[g_3^x]' \rangle$, where $E_T[\cdot]'$ is the reencryption of $E_T[\cdot]$. Now, the talliers check whether the credentials are valid or not and verify whether they are legitimate. For this, they perform the next steps.

- They use the remaining tuples to *check the validity of the credentials*. This process is similar to the original scheme (see Section 2.1). In case of several elections, however, the registrars have to use their new key $\langle R = g_3^{y'}, \widehat{R} = y' \rangle$ generated for the current election;
- Let LV be a list containing the approved tuples (i.e. tuples with valid credentials), the talliers send LV to a mix net and obtain LV'. They also send the ciphertexts $E_T[A]$ of L_2 to a mix net and obtain L_2';
- By means of Weber et al.'s technique, they cooperate to compute a random value r and raise all values $E_T[A]$ of LV' to r as well as the values of L_2';
- They decrypt the resulting lists and verify whether every plaintext in LV' belongs to an element in L_2'. If a match does not exist, they remove the corresponding tuple from the next step;
- In order to finish this phase, the talliers cooperatively decrypt $E_T[v]'$ of the tuples with legitimate credentials and publish the voting results on the WBB.

4 Conclusion

The voting scheme of ABRTY is coercion-resistant and has linear time complexity. As presented, this solution has drawbacks that can limit its use. In this paper, we presented solutions for these drawbacks. In addition, we introduced a new version of ABRTY's scheme which uses these solutions. The new version allows for the revocation of credentials, and identifies valid (but illegitimate) credentials. This improves the original scheme as a collusion of malicious registrars cannot succeed when issuing illegitimate credentials and eligible voters do not need to register again before each new election.

Acknowledgements. The first author's work was partially supported by the "Fundação Amazônia Paraense (FAPESPA)" under the project: Rede de Pesquisa em TIC. The second author's work has been supported by the French "Agence Nationale de la Recherche" under the LYRICS ANR-11-INSE-013 Project.

References

1. Acquisti, A.: Receipt-free homomorphic elections and write-in ballots. Cryptology ePrint Archive, Report 2004/105 (2004), http://eprint.iacr.org/
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, Germany. Dagstuhl Seminar Proceedings, vol. 07311. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl (2008)
3. Araújo, R., Ben Rajeb, N., Robbana, R., Traoré, J., Youssfi, S.: Towards practical and secure coercion-resistant electronic elections. In: Heng, S.-H., Wright, R.N., Goi, B.-M. (eds.) CANS 2010. LNCS, vol. 6467, pp. 278–297. Springer, Heidelberg (2010)
4. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
5. Brickell, E.F. (ed.): CRYPTO 1992. LNCS, vol. 740. Springer, Heidelberg (1993)
6. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell (ed.) [5], pp. 89–105
7. Chaum, D., Ryan, P.Y.A., Schneider, S.: A practical voter-verifiable election scheme. In: De Capitani di Vimercati, S., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 118–139. Springer, Heidelberg (2005)
8. Clark, J., Hengartner, U.: Panic passwords: Authenticating under duress. In: Provos, N. (ed.) HotSec. USENIX Association (2008)
9. Clark, J., Hengartner, U.: Selections: Internet voting with over-the-shoulder coercion-resistance. In: Danezis (ed.) [12], pp. 47–61
10. 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 (2008)
11. Clarkson, M.R., Myers, A.C.: Coercion-resistant remote voting using decryption mixes. Workshop on Frontiers in Electronic Elections (2005)
12. Danezis, G. (ed.): FC 2011. LNCS, vol. 7035. Springer, Heidelberg (2012)
13. Essex, A., Clark, J., Hengartner, U.: Cobra: toward concurrent ballot authorization for internet voting. In: Proceedings of the 2012 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections, EVT/WOTE 2012, p. 3. USENIX Association, Berkeley (2012)
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 (2005)
18. Meng, B.: An internet voting protocol with receipt-free and coercion-resistant. In: CIT, pp. 721–726. IEEE Computer Society (2007)

19. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell (ed.) [15], pp. 31–53
20. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
21. Schläpfer, M., Haenni, R., Koenig, R., Spycher, O.: Efficient vote authorization in coercion-resistant internet voting. In: Kiayias, A., Lipmaa, H. (eds.) VoteID 2011. LNCS, vol. 7187, pp. 71–88. Springer, Heidelberg (2012)
22. Schnorr, C.-P.: Efficient signature generation by smart cards. J. Cryptology 4(3), 161–174 (1991)
23. Schweisgut, J.: Coercion-resistant electronic elections with observer. In: Krimmer, R. (ed.) Electronic Voting. LNI, vol. 86, pp. 171–177. GI (2006)
24. Smith, W.: New cryptographic election protocol with best-known theoretical properties. In: Workshop on Frontiers in Electronic Elections (2005)
25. Spycher, O., Koenig, R.E., Haenni, R., Schläpfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In: Danezis (ed.) [12], pp. 182–189
26. Spycher, O., Koenig, R.E., Haenni, R., Schläpfer, M.: Achieving meaningful efficiency in coercion-resistant, verifiable internet voting. In: Kripp, M.J., Volkamer, M., Grimm, R. (eds.) Electronic Voting. LNI, vol. 205, pp. 113–125. GI (2012)
27. Wang, H., Zhang, Y., Feng, D.: Short threshold signature schemes without random oracles. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 297–310. Springer, Heidelberg (2005)
28. 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 (2007)

A A Weakness of Acquisti's Coercion-Resistant Scheme

A number of schemes proposed in the literature aim at satisfying the coercion-resistant property. However, many of them do not accomplish this. In this appendix we show that Acquisti's scheme [1] does not fulfills this property either. In addition, we present another drawback of this scheme.

A.1 An Overview of the Scheme

The solution of Acquisti [1] employs an idea similar to the scheme of JCJ. A voter obtains a valid credential from the authorities and use it when she want to cast a valid vote. A voter under coercion may deceive adversaries by giving them fake credentials. However, in Acquisti's scheme, the voter receives from the authorities encrypted shares of her credential and later on combines her credential along with her vote intention. In the next paragraphs we give a short description of Acquisti's scheme.

In a setup phase, the authorities compute two sets of keys of an asymmetric cryptosystem with homomorphic property (e.g. Paillier cryptosystem [20]).

One set encrypts credentials and has a public key T_C. The other set encrypts votes and contains a public key T_V. In addition, the authorities compute another set of asymmetric keys of a non-homomorphic cryptosystem. Each authority also generates a share v_i of a vote such that the sum of the shares of all authorities is equal to a vote v for a valid candidate. This is performed for all valid candidates. After computing the shares, each authority generates two ciphertexts for each candidate. He encrypts his share v_i with T_C and T_V apart. Each authority then publishes his two ciphertexts on a bulletin board. Let $E_{T_C}[v_i]$ and $E_{T_V}[v_i]$ be the resulting ciphertexts of a share v_i.

Registration Phase. For each voter, each authority generates random numbers as shares of credentials. Let σ_i represents a share produced by an authority A_i. A_i encrypts σ_i with the public key for credentials T_C, signs the resulting ciphertext, and publishes the signed ciphertext on the bulletin board. After that, A_i encrypts the share σ_i with the public key for votes T_V and adds a designated verifier proof [16] that the ciphertext published on the bulletin board and the ciphertext computed using T_V contain the same σ_i. A_i then encrypts the ciphertext and the proof with the voter's public key and sends the resulting ciphertext to the voter. The authority does not publish the encryption with the public key T_V.

Voting Phase. After decrypting the ciphertexts received from the authorities and verifying that the proofs are correct, the voter multiplies all encrypted shares of her credential (via the homomorphic property of the cryptosystem) and obtains $E_{T_V}[\sigma]$. The voter then reads from the bulletin board the encrypted shares $E_{T_V}[v_i]$ corresponding to the candidate she wants to vote for and multiplies them to obtain $E_{T_V}[v]$. Now, the voter multiplies the two resulting ciphertexts (i.e. $E_{T_V}[\sigma]$ and $E_{T_V}[v]$) and obtains $E_{T_V}[\sigma + v]$. To finish the process, the voter encrypts $E_{T_V}[\sigma + v]$ with the public key of the non-homomorphic cryptosystem and then publishes the resulting ciphertext on the bulletin board.

Tallying Phase. When the voting phase ends, the authorities multiply the shares of all valid encrypted credentials published in the registration phase. As before, this is performed via the homomorphic property of the cryptosystem. The result is a list of entire encrypted credentials $E_{T_C}[\sigma]$ that is sent to a mix net. The mix net outputs a list of $E_{T_C}[\sigma]'$, where $E_{T_C}[\cdot]'$ is the reencryption of $E_{T_C}[\cdot]$. The authorities then remove the encryption layer of the ciphertexts (this is performed using the private key of the non-homomorphic cryptosystem) posted by the voters and obtain a list of ciphertexts $E_{T_V}[\sigma + v]$. The authorities send this list to a mix net that outputs a new list containing $E_{T_V}[\sigma + v]'$.

For each candidate v_i, the authorities read from the bulletin board the corresponding encrypted shares $E_{T_C}[v_i]$ and multiply them. After processing the encrypted shares of all candidates, the authorities obtain a list of ciphertexts $E_{T_C}[v]'$ that contains an encrypted vote for each candidate. Now, for each encrypted credential $E_{T_C}[\sigma]'$, the authorities choose an encrypted vote $E_{T_C}[v]'$. The authorities then decrypt the resulting multiplication of $E_{T_C}[\sigma]'$ and $E_{T_C}[v]'$

(i.e. $E_{T_C}[\sigma+v]'$), and decrypt all ciphertexts $E_{T_V}[\sigma+v]'$. If the resulting plaintext of $E_{T_C}[\sigma+v]'$ matches one of the plaintexts of $E_{T_V}[\sigma+v]'$, the credential is valid and the vote is counted. Otherwise, the authorities combine $E_{T_C}[\sigma]'$ with another encrypted vote $E_{T_C}[v]'$ and repeat the process. If no match is found, the same process is performed with another encrypted credential.

A.2 The Weakness

We show now that the scheme of Acquisti is not coercion-resistant. In particular, an adversary can force a voter to reveal a credential and use a strategy to check later on whether the credential he received is valid or not.

Suppose the voter gives the coercer an encrypted credential $E_{T_V}[\sigma']$. The coercer then employs this ciphertext to compute a new ciphertext in such a way that their underlying plaintexts satisfy a specific relation R. In order to perform this, he first reads the encrypted shares of a vote $E_{T_V}[v_i]$ corresponding to a particular candidate and multiplies them to obtain $E_{T_V}[v]$. After that, he multiplies $E_{T_V}[v]$ to $E_{T_V}[\sigma']$ and obtains $E_{T_V}[v+\sigma']$. Now, the coercer defines the relation R. For example, he selects a value t as before and computes $E_{T_V}[v+\sigma']^t = E_{T_V}[t\cdot(v+\sigma')]$. The adversary then encrypts $E_{T_V}[t\cdot(v+\sigma')]$ and $E_{T_V}[v+\sigma']$ apart with the public key of the non-homomorphic cryptosystem and publishes the resulting ciphertexts on the bulletin board.

After the voting period, the coercer waits until the authorities compute the voting results. As explained above, in this process, the authorities first mix the ciphertexts $E_{T_V}[\sigma+v]$ posted by the voters after removing the outer encryption layer. They then mix the valid credentials ciphertexts $E_{T_C}[\sigma]$ after multiplying their encrypted shares $E_{T_C}[\sigma_i]$. After that, they multiply the mixed values $E_{T_C}[\sigma]'$ to $E_{T_C}[v]$ after obtaining the entire encrypted votes from the shares $E_{T_C}[v_i]$; let $E_{T_C}[X]'$ be the mix net output of a ciphertext $E_{T_C}[X]$, for a message X.

The authorities then decrypt $E_{T_V}[\sigma+v]'$ and $E_{T_C}[\sigma+v]'$, and match their plaintexts to verify whether a vote is valid or not. While processing all ciphertexts, the authorities publish the ciphertexts related information on the bulletin board. At the end of this phase, anyone can check which votes belong to the valid credentials. Now, in order to verify the relation R, the adversary reads from the bulletin board a plaintext $\sigma+v$ corresponding to a ciphertext $E_{T_V}[\sigma+v]'$ and computes $t\cdot(v+\sigma)$ from it. After that, he searches on the bulletin board for a plaintext $t\cdot(v+\sigma)$ of $E_{T_V}[t\cdot(v+\sigma)]'$ that matches the value $t\cdot(v+\sigma)$ he just has computed. If a match exist (which means that, with overwhelming probability, he has identified the ballot $\sigma'+v$ he submitted in an encrypted form), the adversary verifies whether $\sigma+v$ of $E_{T_V}[\sigma+v]'$ has a corresponding valid credential $\sigma+v$ of $E_{T_C}[\sigma+v]'$ on the bulletin board. A match now indicates that the credential received by the coercer is valid. If the value $t\cdot(v+\sigma)$ generated by the adversary does not match any plaintext on the bulletin board, he chooses another credential and repeats the process.

A.3 Another Drawback of Acquisti's Solution

In many election scenarios, voters do not need to register again before each new election. That is, once registered, a voter can vote in many forthcoming elections. The solution of Acquisti, however, does not provide this convenience to voters. They have to register before each new election.

Recall that in his scheme, in order to verify whether a ciphertext $E_{T_C}[\sigma + v]$ matches to a ciphertext $E_{T_V}[\sigma + v]$, the authorities need to decrypt these ciphertexts and publish their respective plaintexts on the bulletin board. Although the credentials (σ) as well as the votes (v) are unknown to the voters as they only know these values in an encrypted form, an adversary could exploit plaintexts $\sigma + v$ to retrieve several arbitrary credentials.

Indeed, suppose that the adversary wants to retrieve the credential of an arbitrary voter V. Let σ' be the credential of the adversary and v' the candidate he chooses. In order to identify his vote during the tallying phase, the adversary first computes the ballot $E_{T_V}[v' + \sigma']$ and publish it on the bulletin board. Then, he uses the same technique as described in the previous section. That is, he chooses a random value t, computes $E_{T_V}[t \cdot (v' + \sigma')]$, and sends this ciphertext to the bulletin board. After the authorities decrypt all ciphertexts $E_{T_V}[\sigma + v]$, the coercer can identify his ballot $\sigma' + v'$. Now, he just has to choose another valid ballot for the same candidate v'. Let us denote this valid ballot by $\sigma + v'$. Then, by subtracting $\sigma' + v'$ from $\sigma + v'$, he will obtain $\sigma - \sigma'$. He can then use the public key T_V to encrypt $\sigma - \sigma'$ in order to obtain $E_{T_V}[\sigma - \sigma']$. Since he knows $E_{T_V}[\sigma']$ (because this is his credential), he can retrieve $E_{T_V}[\sigma]$ via the homorphic property of the cryptosystem. To do this, he just has to multiply $E_{T_V}[\sigma - \sigma']$ by $E_{T_V}[\sigma']$. We tacitly assume here that the computation operator on the underlying homomorphic encryption scheme is multiplication.