Remote Electronic Voting can be Efficient, Verifiable and Coercion-Resistant

Roberto Araújo[1], Amira Barki[2,3], Solenn Brunet[2,4], 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, Caen, France
{amira.barki, solenn.brunet, jacques.traore}@orange.com
[3] Sorbonne universités, Université de technologie de Compiègne (UTC), CNRS, UMR 7253 Heudiasyc, Compiègne, France
[4] Université de Rennes 1, Rennes, France

Abstract. The coercion issue in remote electronic voting has always been of particular interest. However, to date, all proposals addressing it either suffer from some shortcomings or are not efficient enough to be used in real world elections. To fill this gap, we propose a new coercion-resistant electronic voting scheme practical for real polls. Our scheme relies on credentials generated thanks to a recent algebraic Message Authentication Code (MAC) scheme due to Chase et al. To enable multiple elections and credentials revocation, we also design a novel sequential aggregate MAC scheme, that is of independent interest. Thanks to it, eligible voters' credentials can be efficiently updated.

1 Introduction

Internet voting offers a better voting experience since voters can cast their votes from their computers or even smartphones. By eliminating the need to visit polling places, it may attract more voters and thus increase voter turnout. In addition, it improves the efficiency for tallying authorities. These benefits motivated countries such as Estonia and Switzerland to adopt it in real world elections. However, it is still not widely spread. This is particularly due to many inherent concerns such as selective DDoS attacks on the election server, malware attacks on the voter client as well as risks entailed by the lack of private polling booths [14]. In this paper, we will mainly focus on the latter concern while assuming that votes will be cast-as-intended. Indeed, adversaries may leverage it to perform coercion or vote-selling attacks. Consequently, electronic voting schemes ought to address this issue that remained a challenge for many years.

To this end, Juels, Catalano and Jakobsson (JCJ) [10] introduced an essential property known as *coercion-resistance*. It considers the different actions that a coercer could undertake: constrain a voter to cast a given vote, force her to reveal her private vote information and subsequently vote on her behalf, or keep her from voting. They also proposed the first coercion-resistant scheme based

on anonymous credentials. To be able to vote, an eligible voter is beforehand provided with a valid credential. Under coercion, she can use a fake credential instead of her valid one. Thereby, she deceives any adversary about her true vote intention as a coercer is unable to distinguish the fake credential from the valid one. Unfortunately, JCJ's scheme was inefficient for large scale voting scenarios as, for N ballots, the complexity of the tallying is in $O(N^2)$.

Related work. To enhance JCJ's voting system, other coercion-resistant schemes were then proposed[5]. AFT [1] was the first proposal to achieve linear time complexity. Nevertheless, it does not support multiple elections. Indeed, at each new election, the voter has to visit the registration place in order to obtain her credential associated to the new poll. To address this drawback, AT [2] proposed a scheme that allows credentials revocation and multiple elections. To issue a new credential, it requires the registration authorities to jointly generate a BBS [3] group signature. Unfortunately, up to now, there is no practical solution to compute such a signature in a distributed manner, which makes AT impractical for real polls. They also proposed a generic technique to identify valid (but illegitimate) voting credentials that a majority of colluding registrars could compute. Although such an event is unlikely, their generic technique also applies to our scheme. Finally, Clark et al. [7] and Spycher et al. [13] proposed two different approaches to tackle the coercion-resistance issue. However, both schemes do not really have linear time complexity. They truly achieve it only if the level of anonymity is lowered. More specifically, a voter's ballot is indistinguishable from a small set of ballots and not from all the received ones.

Contribution. To tackle all these shortcomings, we propose a novel efficient coercion-resistant voting scheme, with linear time complexity, that is suitable and practical for real polls. Our scheme relies on credentials generated based on the recent Algebraic MAC scheme due to Chase et al. [6]. We prove that although a part of our credentials are made publicly known, a coercer is unable to distinguish a valid credential from a fake one. Furthermore, our scheme allows talliers to check credentials validity while being encrypted. To also enable multiple elections and credentials revocation, we propose a new sequential aggregate signature scheme, which is of independent interest. Using it, eligible voters' credentials can be efficiently updated thereby allowing them to vote in new elections. Credentials of voters who are no longer eligible to vote can be revoked as well. Thanks to our improvements, coercion-resistance is obtained almost for free as our scheme is just slightly slower than non coercion-resistant classical mix-net based voting schemes.

2 Preliminaries

In this section, we first introduce our main notation and required conventional cryptographic primitives. Then, we detail building blocks including our new sequential aggregate signature scheme necessary to update voters' credentials.

[5] Due to lack of space, we only mention the most promising coercion-resistant proposals.

2.1 Classical tools

Notation. The notation $x \in_R X$ states that x is chosen uniformly at random from the set X. Besides, $\vec{x}$ and B_i respectively denote the vector $(x_0, x_1, \ldots, x_n)$ and the ith element of the tuple $B = \langle a, b, \ldots, z \rangle$.

Computational Assumptions. The decisional Diffie-Hellman assumption, known as DDH, is defined as follows: given a cyclic group $\mathbb{G} =< h >$ of prime order p, it is hard, given $(h; h^a; h^b; h^c) \in \mathbb{G}^4$, to decide whether $c \stackrel{?}{=} ab$.

ElGamal encryption. The ElGamal cryptosystem is an asymmetric encryption scheme with multiplicative homomorphic property. Let $\mathbb{G}$ be a cyclic group with safe prime order p. The public key pk is defined as $pk = (g, h = g^{sk})$ where g is a generator of $\mathbb{G}$ and $sk \in_R \mathbb{Z}_p^*$ is the corresponding private key. The ElGamal encryption of a message $m \in \mathbb{G}$ using pk is denoted by $E_{pk}[m]$ and equal to $C = (c_1, c_2)$ where $c_1 = g^r$, $c_2 = mh^r$ and $r \in_R \mathbb{Z}_p^*$. The plaintext is then recovered as $m = c_2/c_1^{sk}$. Its multiplicative homomorphic property states that given two ciphertexts $C_1 = (c_1, c_2)$ and $C_2 = (c'_1, c'_2)$ of the messages m_1 and m_2 respectively, one can efficiently compute the ciphertext of the product $m_1 m_2$ of the two original messages as $C' = (c''_1 = c_1 c'_1, c''_2 = c_2 c'_2)$.

Non-Interactive Zero-knowledge Proofs of Knowledge (NIZKPK). Non Interactive Zero-Knowledge Proofs of Knowledge enable a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that she knows some secrets satisfying a given statement without revealing anything else about them. Following the usual notation introduced by Camenisch and Stadler [5], they are denoted by $\pi = \text{PoK}\{\alpha, \beta : \textit{ statements about } \alpha, \beta\}$ where Greek letters correspond to the knowledge of $\mathcal{P}$.

2.2 Algebraic MACs

Message Authentication Codes (MACs) are cryptographic primitives that rely on pseudorandom functions to provide authentication for messages. In these protocols, MAC construction and verification are performed using the same key. Unlike usual constructions, algebraic MACs are based on group operations.

In what follows, we describe the algebraic $\mathsf{MAC}_{\mathsf{GGM}}$ scheme due to Chase et al. [6]. It is a generalization of the algebraic MAC algorithm proposed by Dodis et al. [8] and is proven unforgeable against chosen message and verification attack (UF-CMVA) in the generic group model.

$\mathsf{Setup}(1^k)$ Define a secure cyclic group $\mathbb{G}$ with prime order p of k-bits as well as g and h, two of its generators such that $\log_g h$ is unknown. It outputs $params = (\mathbb{G}, p, g, h)$.

$\mathsf{Keygen}(params)$ Generate a secret key $sk = \vec{x} \in_R \mathbb{Z}_p^{n+1^*}$. Optionally, compute the parameters $(X_1 = h^{x_1}, X_2 = h^{x_2}, \ldots, X_n = h^{x_n})$ denoted by $iparams$ and $C_{x_0} = g^{x_0} h^{\tilde{x}_0}$, a commitment to x_0 where $\tilde{x}_0 \in_R \mathbb{Z}_p^*$.

$\mathsf{MAC}(sk, \vec{m})$ Given a message $\vec{m} \in_R \mathbb{Z}_p^n$, choose $u \in_R \mathbb{G}\backslash\{1\}$ and compute the tag $\sigma = (u, u')$ where $u' = u^{x_0+m_1x_1+m_2x_2+\ldots+m_nx_n}$.

$\mathsf{Verify}(sk, \vec{m}, \sigma)$ Check the correctness of σ *i.e.* $u \neq 1$ and $u^{x_0+m_1x_1+\ldots+m_nx_n} \stackrel{?}{=} u'$.

We show in the following lemma that it is hard for an adversary to decide whether a given triplet $(s', u, u' = u^{x_0+sx_1})$ is a valid MAC on s or not.

Lemma 1. *Under the DDH assumption, it is unfeasible to decide whether* $s' \stackrel{?}{=} s \bmod p$ *from* s', $C_{x_0} = g^{x_0}h^x$, $X_1 = h^{x_1}$, $u = h^b$, $u' = u^{x_0+sx_1}$ *where* s,s',x,x_0,x_1,$b \in_R \mathbb{Z}_p^*$ *and* $g, h \in_R \mathbb{G}$ *two generators.*

Proof. Suppose that we have an oracle deciding whether $s' \stackrel{?}{=} s \bmod p$ given s', $X_1 = h^{x_1}$, $C_{x_0} = g^{x_0}h^x$, $u = h^b$, $u' = u^{x_0+sx_1}$, for $s, s', x, x_0, x_1, b \in_R \mathbb{Z}_p^*$ and $g, h \in_R \mathbb{G}$ two generators. Then, we show how to decide whether $c \stackrel{?}{=} x_1 b \mod p$ given $h, \alpha = h^{x_1}, \beta = h^b$ and $\gamma = h^c$ for $x_1, b, c \in_R \mathbb{Z}_p^*$, hence contradicting the DDH assumption.

The reduction is as follows. Set $C_{x_0} = g^{x_0}h^x$ for $x_0, x \in_R \mathbb{Z}_p^*$, choose $s' \in_R \mathbb{Z}_p^*$ and give $s', C_{x_0}, X_1 = \alpha, u = \beta, u' = u^{x_0}\gamma^{s'}$ to the oracle. We have two cases:
Case 1. If $c = x_1 b \mod p$ then $u' = u^{x_0+s'x_1}$.
Case 2. If $c \neq x_1 b \mod p$ then $c = x_1 b(1 + c')$ for some $c' \neq 0 \mod p$ (since $x_1 \neq 0$ and $b \neq 0$) and $u' = u^{x_0+sx_1}$ with $s = s'(1 + c') \neq s'$ (since $s' \neq 0$).

Therefore, given $s', C_{x_0}, X_1 = \alpha, u = \beta, u'$, the oracle will tell whether $s' \stackrel{?}{=} s$ from which we decide whether $c \stackrel{?}{=} x_1 b$.

In the particular case where $s = 0$, even a computationally unbounded adversary will not be able to figure out whether $u' \stackrel{?}{=} u^{x_0}$. This is due to the fact that the Pedersen's commitment $C_{x_0} = g^{x_0}h^x$ perfectly hides the value x_0.

2.3 Our Sequential Aggregate MAC Scheme

An aggregate signature scheme [4] is a variant of a digital signature scheme that additionally supports aggregation. Indeed, it allows to aggregate n signatures on n distinct messages from n signers into a single compact signature. Along with the n messages, the resulting signature will convince the verifier that the n messages were signed by the n signers. A sequential aggregate signature scheme [11] is a particular type of aggregate signature schemes. Indeed, the final signature is created sequentially with each signer signing the aggregate signature in turn. Based on the $\mathsf{MAC}_{\mathsf{GGM}}$ due to Chase et al, we design a new sequential aggregate signature (MAC) scheme which supports n signers with n different messages. In the case of two signers S_1 and S_2, it works as follows:

$\mathsf{Setup}(1^k)$ Create the system public parameters $param = (\mathbb{G}, p, g, h)$ as defined in Section 2.2.

$\mathsf{KeyGeneration}(params)$ Generate the secret key $sk_1 = (x_0, x_1)$ of the first signer $\mathcal{S}_1$ and $sk_2 = x_2$ of the second signer $\mathcal{S}_2$. The corresponding public parameters are respectively $C_{x_0} = g^{x_0}h^x$ where $x \in_R \mathbb{Z}_p^*$, $X_1 = h^{x_1}$ and $X_2 = h^{x_2}$.

$\mathsf{Signing}(params, \mathcal{S}_1(sk_1, m_1), \mathcal{S}_2(sk_2, m_2))$ Produce an aggregate signature on messages m_1 and m_2 sequentially by $\mathcal{S}_1$ and $\mathcal{S}_2$. First, $\mathcal{S}_1$ generates the signature $\sigma_1 = (u, u')$ on the message m_1 where $u' = u^{x_0 + m_1 x_1}$. Then, $\mathcal{S}_2$ can generate $\sigma_2 = (w = u^t, w' = (u' u^{m_2 x_2})^t)$ a sequential aggregate signature on both m_1 and m_2 where $t \in_R \mathbb{Z}_p$.

$\mathsf{Verification}(params, \sigma_2, m_1, m_2, sk)$ Verify that σ_2 is the aggregate of the signatures of $\mathcal{S}_1$ on m_1 and $\mathcal{S}_2$ on m_2 *i.e.* $u \neq 1$ and $w' \stackrel{?}{=} w^{x_0 + m_1 x_1 + m_2 x_2}$.

Theorem 1. *Our sequential aggregate signature scheme is* existentially unforgeable under chosen message attacks *(EUF-CMA) under the assumption that* $\mathsf{MAC}_{\mathsf{GGM}}$ *is UF-CMVA.*

Proof. Owing to space limitations, the proofs will be detailed in an extended version. It is, however, worth mentioning that the EUF-CMA proof is similar to the one provided in [12].

We will subsequently use the designed sequential aggregate MAC scheme in our voting scheme to update voter's credentials thereby enabling multiple elections and credentials revocation. m_2 and x_2 will be respectively set to the new election identifier and the associated secret key.

3 A MAC based Coercion Resistant Voting Scheme

In this section, we first provide an overview of our coercion-resistant voting scheme then, detail it while explaining how it enables multiple elections and credentials revocation.

3.1 An Overview of the Scheme

Our coercion-resistant voting scheme consists of five main phases. During the *setup* phase, key material as well as election parameters are cooperatively generated by a set of authorities. The public parameters are then published on a Web Bulletin Board (WBB). To be able to vote, an eligible voter must register through a *registration* phase. After proving her identity, she receives a unique and valid credential that depends on a secret s only known by the voter. The credential is issued by the registration authorities and made publicly available. Later, during *voting* phase, the voter uses her credential and the secret s to generate a ballot that she sends via an anonymous channel. It contains her credential randomized, the ciphertext of her vote as well as a set of NIZKPs proving the validity of the ballot. If the voter is under coercion, she can cast a fake ballot using an invalid secret s' without the adversary being able to distinguish it from a valid one. Before tallying votes, a *pre-verification* phase is carried out to remove both erroneous ballots and duplicate votes. Once done, the tallying authorities may perform the *tallying* phase. To do so, they first send the remaining ballots to a verifiable mix net and then anonymously identify the valid votes, *i.e.* votes published with valid credentials. Finally, they cooperatively decrypt the associated ciphertexts to recover votes and publish results on the WBB.

3.2 Our Novel Coercion-resistant Voting Scheme

Our voting scheme, which assumes a bulletin board communication model, involves as participants a set of registration authorities known as *registrars*, a set of tallying authorities called *talliers*, and a set of *voters*. For security reasons, the roles of both registrars and talliers are distributed among a large group of authorities. We describe our proposal as follows:

Setup Phase. Let O be the set of eligible options (candidates) and $v \in O$ a vote for a candidate. Let $\mathbb{G}$ be a cyclic group with a prime order p and $o \in \mathbb{G}$ be a public generator selected for this election. The talliers share the threshold ElGamal key pair $(T, \widehat{T})$. As for the registrars, they jointly select and share a secret key $sk = (x_0, x_1) \in_R \mathbb{Z}_p^{2*}$ associated to the public values $(C_{x_0}, X_1 = h^{x_1})$ where $C_{x_0} = g^{x_0}h^x$ such that $x \in_R \mathbb{Z}_p^*$. This key is also shared among talliers.

Registration Phase. Once her eligibility proved, the voter obtains a unique and valid voting credential σ. Indeed, by cooperatively choosing $s \in_R \mathbb{Z}_p$ and $u \in_R \mathbb{G}\backslash\{1\}$, the registrars jointly compute $\sigma = (u, u')$ where $u' = u^{x_0 + sx_1}$. It is then provided, through an untappable channel, to the voter along with the secret value s and a Designated Verifiable Proof[6] [9] that σ is a valid credential on s. Concurrently, the credential σ is stored in the database DB, which contains all the valid credentials, while s is kept as a secret only known by the voter. Thereby, in case of coercion, the voter would deceive the coercer by revealing her credential with a fake value s' without the coercer noticing it. Indeed, under the DDH assumption, the coercer cannot decide whether s' is valid with respect to the voter's credential σ or not (see lemma 1). The generic technique proposed in [2] can be subsequently used to detect any vote cast with a valid but illegitimate credential computed by a set of malicious colluding registrars.

Voting Phase (First Election). To vote in a first election, the voter first randomizes the received credential σ to generate $\sigma_r = (u^r, u'^r) = (w, w')$ where $r \in_R \mathbb{Z}_p$. Then, she chooses her candidate $v \in O$ and casts her vote that consists of the ballot $B = \langle E_T[v], w, w', E_T[w^s], o^s, P\rangle$ where P is a set of NIZKPs ensuring that the ballot is well formed. In particular, P includes both $\pi_1 = PoK\{\alpha : B_4 = E_T[w^\alpha] \wedge B_5 = o^\alpha\}$ related to the knowledge of s and $\pi_2 = PoK\{\beta : B_1 = E_T[\beta] \wedge \beta \in O\}$ proving that v belongs to the set O.

Pre-Verification Phase. This phase aims to verify votes posted on the WBB. Talliers should perform it before the tallying phase detailed later on. It is worth mentioning that, during this phase, ballots with invalid credentials are not yet discarded. Hereinafter, we describe the four steps of this phase.

[6] The DVP proof can only convince the corresponding voter and nobody else. So, it is useless in case of coercion and even for vote-selling.

1. *Verifying proofs.* For each posted ballot, the proofs P are verified to remove ballots with invalid proofs.
2. *Removing duplicates.* By comparing all o^s values, duplicates votes (*i.e.* ballots published using the same secret s) are removed. The policy, in this case, could be to keep the last one.
3. *Reconstruction of the credential.* For each ballot, the ElGamal ciphertext $E_T[w]$ of w is cooperatively computed. Using ElGamal homomorphic property, the ciphertexts $E_T[w^{x_0}]$ and $E_T[(w^s)^{x_1}]$ are jointly obtained thanks to the shared secret values x_0 and x_1 as well as $E_T[w]$ and $E_T[w^s]$. Thereby, the talliers can compute $E_T[w^{x_0+sx_1}] = E_T[w^{x_0}] \cdot E_T[w^{sx_1}]$. By dividing the ciphertext second component by w', they obtain $C = E_T[w^{x_0+sx_1}]/w'$. If the credential $\sigma_r = (w, w')$ is valid, C should be equal to $E_T[1]$, a ciphertext of 1.
4. *PET pre-test.* In this last step, a Plaintext Equivalence Test (PET) is performed on credentials. To this end, C is cooperatively raised to a random value $\alpha \in_R \mathbb{Z}_p$. For a valid credential σ, $D = C^\alpha$ should be equal to $E_T[1^\alpha] = E_T[1]$. Note that D is still kept encrypted to prevent any information leakage especially in case of coercion.

Tallying Phase. To compute election results, the talliers perform three steps:

1. *Mixing tuples.* The tuples $\langle D, E_T[v]\rangle$ that succeeded all pre-verifications are sent to a verifiable mix net. The output is then published on the WBB.
2. *Identifying valid votes.* For each tuple, the ciphertext D is jointly decrypted. If the plaintext is equal to 1, the credential σ_r and the associated ballot are considered as valid. Otherwise, the ballot is said invalid and is thus discarded.
3. *Decrypting and counting votes.* Finally, for each valid ballot, $E_T[v]$ is cooperatively decrypted in order to count the votes. The obtained results are then published on the WBB.

Theorem 2. *Our voting scheme satisfies the* eligibility[7] *requirement under the assumption that* $\mathsf{MAC}_{\mathsf{GGM}}$ *is UF-CMVA secure and the* coercion-resistance[8] *requirement, in the random oracle model, under the DDH assumption.*

Proof (sketch). Owing to space limitations, the proofs will be detailed in an extended version. Intuitively, eligibility follows from the unforgeability of the $\mathsf{MAC}_{\mathsf{GGM}}$ and the removal of duplicates in step 2. Therefore, only one vote per credential (valid or fake) will be processed during the tallying phase. Coercion-resistance follows from the fact that a coercer cannot decide, under the DDH assumption, whether a credential is valid or not (see Lemma 1) or trace a ballot during the tallying phase (owing to the use of Mix-nets and PET that are secure under the DDH assumption).

[7] The eligibility requirement informally states that only eligible voters can cast the votes and that every voter can cast only one vote.

[8] As defined by JCJ [10] : A voter can deceive the coercer about her true vote intention by making him believe that she behaved as instructed while it is not the case.

Universal Verifiability: We would also like to stress that every step of the tallying phase is publicly verifiable. Thus, anyone can check that the election outcome corresponds to the ballots published on the WBB (Universal Verifiability) and in particular, that only invalid ballots containing invalid credentials have been discarded.

For every new election or in the case where some voters are no longer eligible, the authorities should be able to update eligible voters' credentials without requiring them to register again. To this end, we design the following scheme that relies on our sequential aggregate signature scheme introduced in Section 2.3.

Multiple Elections and Credentials Revocation. For every new election, the registrars generate both a specific election identifier and a new pair of keys. For the ith election, this pair is defined as $(x_i, X_i = h^{x_i})$ where $x_i \in_R \mathbb{Z}_p$ is shared among registrars and talliers. Hereinafter, we detail the case of a second election identified by e_I and where the new key pair is $(x_2, X_2 = h^{x_2})$.

For each initial credential $\sigma = (u, u') \in DB$ belonging to an eligible voter, the registrars jointly select a random value $t \in_R \mathbb{Z}_p$, compute $\sigma_2 = (u^t, (u'u^{e_I x_2})^t) = (w, w' = w^{x_0 + s x_1 + e_I x_2})$ and update DB. Then, the new database is published to enable eligible voters to learn their new credential. These changes are irrelevant except for the pre-verification phase whose third step requires these modifications:

- *Reconstruction of the credential.* First, the talliers cooperatively encrypt w to get $E_T[w]$. Then, as previously and thanks to ElGamal homomorphic property, they jointly compute the three ciphertexts: $E_T[w^{x_0}]$, $E_T[w^{s x_1}]$ and $E_T[w^{e_I x_2}]$ using e_I, $E_T[w]$ and $E_T[w^s]$ as well as their shared secret keys x_0, x_1 and x_2. Thereby, the talliers can compute $E_T[w^{x_0}] \cdot E_T[w^{s x_1}] \cdot E_T[w^{e_I x_2}] = E_T[w^{x_0 + s x_1 + e_I x_2}]$. By dividing the ciphertext second component by w', they obtain $C = E_T[w^{x_0 + s x_1 + e_I x_2}]/w'$. If the associated credential is valid, C should be equal to $E_T[1]$, a ciphertext of 1.

4 Conclusion

We proposed a new efficient coercion-resistant voting scheme that enables credentials revocation as well as multiple elections without requiring voters to visit the registration place again. This is achieved through the design of a new sequential aggregate MAC scheme based on Chase et al. Algebraic MAC scheme.

References

1. 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. pp. 330–342. Schloss Dagstuhl, Germany (2007)

2. Araújo, R., Traoré, J.: A practical coercion resistant voting scheme revisited. In: Heather, J., Schneider, S., Teague, V. (eds.) Vote-ID 2013. LNCS, vol. 7985, pp. 193–209. Springer, Heidelberg (2013)
3. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) Advances in Cryptology - CRYPTO 2004, LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
4. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) Advances in Cryptology - EUROCRYPT 2003, LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
5. Camenisch, J., Stadler, M.: Proof systems for general statements about discrete logarithms. Tech. rep. (1997)
6. Chase, M., Meiklejohn, S., Zaverucha, G.: Algebraic macs and keyed-verification anonymous credentials. In: Proceedings of the 2014 ACM SIGSAC CCS. pp. 1205–1216. CCS '14, ACM, New York, NY, USA (2014)
7. Clark, J., Hengartner, U.: Selections: Internet voting with over-the-shoulder coercion-resistance. In: FC 2011. pp. 47–61 (2011)
8. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: Advances in Cryptology - EUROCRYPT 2012. LNCS, vol. 7237, pp. 355–374. Springer, Heidelberg (2012)
9. Jakobsson, M., Sako, K., Impagliazzo, R.: Advances in Cryptology — EUROCRYPT '96: International Conference on the Theory and Application of Cryptographic Techniques Saragossa, chap. Designated Verifier Proofs and Their Applications, pp. 143–154. Springer Berlin Heidelberg (1996)
10. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Atluri, V., di Vimercati, S.D.C., Dingledine, R. (eds.) WPES. pp. 61–70. ACM (2005)
11. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology - EUROCRYPT 2004, LNCS, vol. 3027, pp. 74–90. Springer, Heidelberg (2004)
12. Pointcheval, D., Sanders, O.: Short randomizable signatures. Cryptology ePrint Archive, Report 2015/525 (2015)
13. Spycher, O., Koenig, R.E., Haenni, R., Schläpfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In: FC 2011. pp. 182–189 (2011)
14. US Vote Foundation: End-to-end verifiable internet voting. In: The future of voting, Expert Statement (2015), `https://www.usvotefoundation.org/sites/default/files/E2EVIV_expert_statements.pdf`