Coercion-Resistant Voting in Linear Time via Fully Homomorphic Encryption

Towards a Quantum-Safe Scheme

Peter B. Rønne[1], Arash Atashpendar[1], Kristian Gjøsteen[2], and Peter Y. A. Ryan[1]

[1] SnT, University of Luxembourg, Luxembourg
{peter.roenne,arash.atashpendar.peter.ryan}@uni.lu
[2] Norwegian University of Science and Technology, NTNU, Norway
kristian.gjosteen@ntnu.no

Abstract. We present an approach for performing the tallying work in the coercion-resistant JCJ voting protocol, introduced by Juels, Catalano, and Jakobsson, in linear time using fully homomorphic encryption (FHE). The suggested enhancement also paves the path towards making JCJ quantum-resistant, while leaving the underlying structure of JCJ intact. The pairwise comparison-based approach of JCJ using plaintext equivalence tests leads to a quadratic blow-up in the number of votes, which makes the tallying process rather impractical in realistic settings with a large number of voters. We show how the removal of invalid votes can be done in linear time via a solution based on recent advances in various FHE primitives such as hashing, zero-knowledge proofs of correct decryption, verifiable shuffles and threshold FHE. We conclude by touching upon some of the advantages and challenges of such an approach, followed by a discussion of further security and post-quantum considerations.

1 Introduction

Over the past few decades, we have witnessed significant advances in cryptographic voting protocols. Yet, despite all the progress, see e.g., [1], secure e-voting is still faced with a plethora of challenges and open questions, which largely arise as a result of the interplay between intricate properties such as vote privacy, individual and universal verifiability, receipt-freeness, and a notoriously difficult requirement, namely that of coercion-resistance. Coercion-resistance can be viewed as a stronger form of privacy that should hold even against an adversary who may instruct honest parties to carry out certain computations while potentially even requiring that they reveal secrets in order to verify their behavior and ensure compliance. This property is typically enforced by providing honest parties with a mechanism that allows them to either deceive the coercer or to deny having performed a particular action. Due to limited space, we do not elaborate on the long series of works in this area and instead refer the reader to [21,12,22,11] and references therein for more details.

This is a copy of the author preprint. The final authenticated version will appear in the proceedings of Financial Cryptography and Data Security 2019, published by Springer.

Since the breakthrough work of Gentry [15] on fully homomorphic encryption (FHE), there has been a surge of interest in this line of research that remains very active to this day, with a series of recent advances including, but not limited to, a homomorphic evaluation of AES [16]. Although the use of additively or multiplicatively homomorphic cryptosystems is common place in the e-voting literature, the relevance of FHE for potentially quantum-safe secure e-voting, with better voter verifiability, was only recently discussed by Gjøsteen and Strand [17]. In our work, instead of designing an FHE-based protocol from scratch, we apply the machinery of FHE to a well-known, classical voting scheme, in order to improve its time complexity and to replace its reliance on the hardness assumption of solving the discrete logarithm problem with a quantum-resistant solution, namely lattice-based cryptography. So far, no efficient quantum algorithms capable of breaking lattice-based FHE schemes have been discovered.

Although constructions with varying degrees of coercion-resistance do exist, the voting protocol introduced by Juels, Catalano, and Jakobsson [21], often referred to as the *JCJ protocol*, is among the most well-known solutions in the context of coercion-resistant voting schemes. JCJ provides a reasonable level of coercion-resistance using a voter credential faking mechanism, and it was arguably the first proposal with a formal definition of coercion-resistance. However, JCJ suffers from a complexity problem due to the weeding steps in its tallying phase, which are required for eliminating invalid votes and duplicates. The exhaustive, comparison-based approach of JCJ using plaintext equivalence tests (PET) [19] leads to a quadratic blow-up in the number of votes, which makes the tallying process rather impractical in realistic settings with a large number of voters or in the face of ballot-box stuffing. For instance, in the Civitas voting system [10] based on JCJ, voters are grouped into blocks or virtual precincts to reduce the tallying time.

Here we propose an enhancement of the JCJ protocol aimed at performing its tallying work in linear time, based on an approach that incorporates primitives from the realm of fully homomorphic encryption (FHE), which also paves the path towards making JCJ quantum-safe.

In Sect. 2, we describe the JCJ protocol and cover some related work. Next, in Sect. 3, we show how the weeding of "bad" votes can be done in linear time, with minimal change to JCJ, via an approach based on recent advances in various FHE primitives such as hashing, zero-knowledge (ZK) proofs of correct decryption, verifiable shuffles and threshold FHE. We also touch upon some of the advantages and remaining challenges of such an approach in Sect. 3.2 and in Sect. 4, we discuss further security and post-quantum considerations.

2 The JCJ Model and Voting Protocol in a Nutshell

Cryptographic Building Blocks. JCJ relies on a modified version of ElGamal, a threshold public-key cryptosystem with re-encryption, secure under the hardness assumption of the Decisional Diffie-Hellman (DDH) problem in a multiplicative cyclic group $\mathcal{G}$ of order q. A ciphertext on message $m \in \mathcal{G}$ has the form $(\alpha, \beta, \gamma) =$

This is a copy of the author preprint. The final authenticated version will appear in the proceedings of Financial Cryptography and Data Security 2019, published by Springer.

(mh^r, g_1^r, g_2^r) for $r \in_U \mathbb{Z}_q$, with (g_1, g_2, h) being the public key where $g_1, g_2, h \in \mathcal{G}$, and the secret key consists of $x_1, x_2 \in \mathbb{Z}_q$ such that $h = g_1^{x_1} g_2^{x_2}$. The construction allows easy sharing of the secret key in a threshold way. The weeding steps make use of a plaintext equivalence test (PET), which is carried out by the secret key holders and takes as input two ciphertexts and outputs 1 if the underlying plaintexts are equal, and 0 otherwise. The PET produces publicly verifiable evidence with negligible information leakage about plaintexts. Finally, JCJ uses non-interactive zero-knowledge (NIZK) proofs and mix-nets, which are aimed at randomly and secretly permuting and re-encrypting input ciphertexts such that output ciphertexts cannot be traced back to their corresponding ciphertexts. Throughout, it is assumed that the computations of the talliers and registrars are done in a joint, distributed threshold manner. We use $\in_U$ to denote an element that is sampled uniformly at random.

Agents. JCJ mainly consists of three sets of agents, described as follows.

1. **Registrars**: A set $\mathcal{R} = \{R_1, R_2, \ldots, R_{n_R}\}$ of n_R entities in charge of jointly generating and distributing credentials to voters.
2. **Talliers**: A set $\mathcal{T} = \{T_1, T_2, \ldots, T_{n_T}\}$ of *authorities* in charge of processing ballots, jointly counting the votes and publishing the final tally.
3. **Voters**: A set of n_V voters, $\mathcal{V} = \{V_1, V_2, \ldots, V_{n_V}\}$, participating in an election, where each voter V_i is publicly identified by an index i.

Bulletin Board and Candidate Slate. A *bulletin board*, denoted by $\mathfrak{BB}$, is an abstraction representing a publicly accessible, append-only, but otherwise immutable board, meaning that participants can only add entries to $\mathfrak{BB}$ without overwriting or erasing existing items. A *candidate slate*, $\boldsymbol{C}$, is an ordered set of n_C distinct identifiers $\{c_1, c_2, \ldots, c_{n_C}\}$ capturing voter choices. A *tally* is defined under slate $\boldsymbol{C}$, as a vector $\boldsymbol{X} = \{x_1, x_2, \ldots, x_{n_C}\}$ of n_C positive integers, where each x_j indicates the number of votes cast for choice c_j.

Assumptions for Coercion-Resistance. No threshold set of agents in $\mathcal{T}$ should be corrupted, otherwise privacy is lost. In the registration phase, it is assumed that the distribution of voter credentials is done over an untappable channel and that no registration transcripts can be obtained, assuming that secure erasure is possible. Cast votes are transmitted via anonymous channels, which is a basic requirement for ruling out forced-abstention attacks.

2.1 The JCJ Protocol

Setup and Registration. The key pairs $(sk_{\mathcal{R}}, pk_{\mathcal{R}})$ and $(sk_{\mathcal{T}}, pk_{\mathcal{T}})$ are generated in a trustworthy manner, and the public keys, i.e., $pk_{\mathcal{T}}$ and $pk_{\mathcal{R}}$, are published with other public system parameters. The registrars $\mathcal{R}$ generate and transmit to eligible voter V_i a random string $\sigma_i \in_U \mathcal{G}$ that serves as the credential of the voter. $\mathcal{R}$ adds an encryption of σ_i, $S_i = E_{pk_{\mathcal{T}}}(\sigma_i)$, to the voter roll $\boldsymbol{L}$, which is maintained on the bulletin board $\mathfrak{BB}$ and digitally signed by $\mathcal{R}$.

Voting. An integrity-protected candidate slate $\boldsymbol{C}$ containing the names and unique identifiers in $\mathcal{G}$ for n_C candidates, along with a unique, random election identifier ϵ are published by the authorities. Voter V_i generates a ballot in the form of a variant of ElGamal ciphertexts (E_1, E_2), for candidate choice c_j and voter credential σ_i, respectively, e.g., for $a_1, a_2 \in_U \mathbb{Z}_q$, we have $E_1 = (g_1^{a_1}, g_2^{a_1}, c_j h^{a_1})$ and $E_2 = (g_1^{a_2}, g_2^{a_2}, \sigma_i h^{a_2})$. V_i computes NIZK proofs of knowledge and correctness of σ_i and $c_j \in \boldsymbol{C}$, collectively denoted by P_f. These ensure non-malleability of ballots, also across elections by including ϵ in the hash of the Fiar-Shamir heuristic. V_i posts $B_i = (E_1, E_2, P_f)$ to $\mathfrak{BB}$ via an anonymous channel.

Tallying. In order to compute the tally, duplicate votes and those with invalid credentials will have to be removed. The complexity problem crops up in steps 2 and 4 such that given n votes, the tallying work has a time complexity of $\mathcal{O}(n^2)$. To tally the ballots posted to $\mathfrak{BB}$, the authority $\mathcal{T}$ performs the following steps: 1. $\mathcal{T}$ verifies all proofs on $\mathfrak{BB}$ and discards any ballots with invalid proofs. Let $\boldsymbol{A_1}$ and $\boldsymbol{B_1}$ denote the list of remaining E_1 candidate choice ciphertexts, and E_2 credential ciphertexts, respectively. 2. $\mathcal{T}$ performs pairwise PETs on all ciphertexts in $\boldsymbol{B_1}$ and removes duplicates according to some fixed criterion such as the order of postings to $\mathfrak{BB}$. For every element removed from $\boldsymbol{B_1}$, the corresponding element with the same index is also removed from $\boldsymbol{A_1}$, resulting in the "weeded" vectors $\boldsymbol{B_1'}$ and $\boldsymbol{A_1'}$. 3. $\mathcal{T}$ applies a mix-net to $\boldsymbol{A_1'}$ and $\boldsymbol{B_1'}$ using the same, secret permutation, resulting in the lists of ciphertexts $\boldsymbol{A_2}$ and $\boldsymbol{B_2}$. 4. $\mathcal{T}$ applies a mix-net to the encrypted list $\boldsymbol{L}$ of credentials from the voter roll and then compares each ciphertext of $\boldsymbol{B_2}$ to the ciphertexts of $\boldsymbol{L}$ using a PET. $\mathcal{T}$ keeps a vector $\boldsymbol{A_3}$ of all ciphertexts of $\boldsymbol{A_2}$ for which the corresponding elements of $\boldsymbol{B_2}$ match an element of $\boldsymbol{L}$, thus achieving the weeding of ballots with invalid voter credentials. 5. $\mathcal{T}$ decrypts all ciphertexts in $\mathbf{A_3}$ and tallies the final result.

Properties. Vote **privacy** is maintained as long as neither a threshold set of talliers nor all the mixing servers are corrupted. A colluding majority of talliers can obviously decrypt everything and colluding mixing authorities could trace votes back to $\boldsymbol{L}$. Regarding **correctness**, voters can refer to $\mathfrak{BB}$ to verify that their vote has been recorded as intended and that the tally is computed correctly. Similar attacks become possible in case of collusion by a majority of authorities. As for **verifiability**, anyone can refer to $\mathfrak{BB}$, P_f and $\boldsymbol{L}$ to verify the correctness of the tally produced by $\mathcal{T}$. The **coercion-resistance** provided by JCJ essentially comes from keeping voter credentials hidden throughout the election. A coerced voter can then choose a random fake credential σ' to cast a fake vote and present it as their real vote. Any vote cast with the fake credential will not be counted, and the voter can anonymously cast their real vote using their real credential.

2.2 Related Work

We focus on the most closely-related works on improving the efficiency problem of the tallying work in JCJ. Smith [25] and Weber et al. [30,29] follow a similar approach in that they do away with comparisons using PETs, and instead, they

This is a copy of the author preprint. The final authenticated version will appear in the proceedings of Financial Cryptography and Data Security 2019, published by Springer.

raise the credentials to a jointly $\mathcal{T}$-shared secret value and store these blinded terms in a hash table such that collisions can be found in linear time. The use of a single exponent means that a coercer can test if the voter has provided them with a fake or a real credential by submitting a ballot with the given credential and another with the credential raised to a known random value. In [4,5], Araujo et al. move away from comparing entries in $\boldsymbol{L}$ with terms in the cast ballots to a setting in which duplicates are publicly identifiable and a majority of talliers use their private keys to identify legitimate votes, and in [3] the authors use algebraic MACs. Spycher et al. [26] use the same solution proposed by Smith and Weber to remove duplicates and apply targeted PETs only to terms in $\boldsymbol{L}$ and $\boldsymbol{A}$, identified via additional information provided by voters linking their vote to the right entry in $\boldsymbol{L}$. In [18], publicly auditable conditional blind signatures are used to achieve coercion-resistance in linear time using a FOO-like [14] architecture, the downsides being the need for extra authorization requests for participation privacy and a double use of anonymous channels.

3 JCJ in Linear Time via Fully Homomorphic Encryption

Our proposal revolves around replacing the original cryptosystem of JCJ with a fully homomorphic one, thus allowing us to preserve the original design of JCJ. The main idea is to homomorphically evaluate hashes of the underlying plaintext of the FHE-encrypted voter credentials, perform FHE-decryption and post the hash values of the credentials to the bulletin board $\mathfrak{BB}$. Now the elimination of invalid and duplicate entries can be done in linear time by using a hash table.

FHE Primitives. Constrained by limited space, we only enumerate the cryptographic primitives that will be required for the enhancement suggested below and refer the reader to the cited sources for further details. Let $\mathcal{E}_{pk}(m)$ denote an FHE-encryption of a message $m \in \{0,1\}^n$ under the public key pk. At its core, for $b_0, b_1 \in \{0,1\}$, given $\mathcal{E}_{pk}(b_0)$ and $\mathcal{E}_{pk}(b_1)$, FHE allows us to compute $\mathcal{E}_{pk}(b_0 \oplus b_1)$ and $\mathcal{E}_{pk}(b_0 \cdot b_1)$ by working over ciphertexts alone, without having access to the secret key, thus enabling the homomorphic evaluation of any boolean circuit, i.e., computing $\mathcal{E}_{pk}(f(m))$ from $\mathcal{E}_{pk}(m)$ for any computable function f. We make use of FHE [15,7], fully homomorphic hashing [13], zero-knowledge proofs of correct decryption for FHE ciphertexts [8], verifiable shuffles [27] and threshold FHE [6], see Sect. 3.2 for more details on open questions and the state-of-the-art.

3.1 Enhancing JCJ with FHE and Weeding in Linear Time

We now describe how FHE primitives can be incorporated into JCJ while inducing minimal change in the original protocol. We assume threshold FHE throughout.

Setup and Registration. The setup and registration phases remain unchanged w.r.t. JCJ, except that $\mathcal{R}$ now adds an FHE-encryption of σ_i, $S_i = \mathcal{E}_{pk_{\mathcal{T}}}(\sigma_i)$, to the voter roll $\boldsymbol{L}$. We adopt the same assumptions mentioned earlier in Sect. 2.

Voting. Instead of using ElGamal encryption, the credentials posted on the $\mathfrak{BB}$ are encrypted under some FHE scheme, say BGV [7], with a key pair (pk, sk). Each voter V_i adds $\mathcal{E}_{pk_{\mathcal{T}}}(\sigma_i)$, along with the required NIZK proofs, to $\mathfrak{BB}$.

Tallying. The tallying phase remains largely the same except that for removing duplicates and invalid votes, we leverage our use of FHE to carry out simple equality tests between hash digests of credentials. Since the concealed credentials are now stored in FHE ciphertexts, we can process them using an FHE hashing circuit. More precisely, for a jointly created $\mathcal{T}$-shared key k, published under encryption $\mathcal{E}_{pk}(k)$, the credentials σ_i contained in the FHE-encrypted terms $\mathcal{E}_{pk}(\sigma_i)$ are homomorphically hashed (see [13] by Fiore, Gennaro and Pastro and [9] by Catalano et al.), under key k resulting in $\mathcal{E}_{pk}(h_k(\sigma_i))$, such that upon decryption we obtain $h_k(\sigma_i)$. A ZK proof of correct decryption is also posted to $\mathfrak{BB}$ for verifiability, see [8] by Carr et al. for an approach to this. Once the hash values of the credentials are posted on the $\mathfrak{BB}$, the weeding of duplicates can be done in $\mathcal{O}(n)$ using a simple hash table look-up, i.e., iterate, hash and check for collision in constant time, thus an overall linear-time complexity in the number of votes. Next, the registered credentials and the submitted vote/credential pairs are mixed [27] and the homomorphic hashing procedure is carried out again using a new secret key on all credential ciphertexts. Comparing the hashed registered credentials with those from the cast ballots allows us to remove invalid votes in $\mathcal{O}(n)$. Finally, the remaining valid votes are verifiably decrypted.

3.2 Advantages, Potential Pitfalls and Open Questions

Apart from the linear-time weeding algorithm, as already pointed out by Gjøsteen and Strand in [17], in addition to being a novel application of FHE to secure e-voting, obtaining better voter verifiability and a scheme believed to be quantum-resistant are among the noteworthy benefits of such an approach. Clearly, in terms of real world FHE implementations, the state-of-the-art still suffers from efficiency issues. However, some significant progress has already been made in this area, e.g., the homomorphic evaluation of AES [16] or block ciphers designed for homomorphic evaluation [2]. Moreover, it should be pointed out that some of the needed primitives, e.g., turning ZK proofs of correct decryption for FHE [8,23] into NIZK proofs, are still not satisfactory and remain the subject of ongoing research and future improvements.

4 Further Security Remarks

A security analysis aimed at providing proofs of security for various properties such as correctness, verifiability and coercion-resistance will remain future work. One possibility would be to investigate whether the required security properties in our enhanced variant of JCJ hold against classical adversaries, under the same oracle access assumptions for mixing, PETs, threshold decryption and hashing. Post-quantum security will have to be proved in the quantum random oracle model.

This is a copy of the author preprint. The final authenticated version will appear in the proceedings of Financial Cryptography and Data Security 2019, published by Springer.

Eligibility Verifiability. Assuming a majority of colluding authorities, apart from a compromise of vote privacy, another, perhaps more damaging problem with JCJ and its improved variants is that of *eligibility verifiability*. A colluding majority would be able to retrieve voter credentials and submit valid votes for non-participating voters, i.e., perform ballot stuffing. A solution in [24] suggests performing the registration phase in such a way that only the voter would know the discrete logarithm of their credential. Votes are then cast with an anonymous signature in the form of a ZK proof of knowledge of the discrete logarithm of the encrypted credential, thus preventing ballot stuffing. A similar approach could be used here, with the potential downside of having inefficient proofs and a discrete logarithm hardness assumption, thus not being quantum secure.

Post-Quantum Considerations. For a relaxation of the trustworthiness assumption of $\mathcal{R}$, without assuming secure erasure, quantum-resistant designated verifier proofs [28,20] could replace the classical ones suggested in the original JCJ [21]. To obtain post-quantum security for eligibility verifiability, future research will investigate the use of a quantum-resistant signature scheme that can be evaluated under FHE to preserve ballot anonymity. As a naive, but illustrative example that is one-time only and non-distributive, consider that the voter creates their credential as $\sigma_i = h(x)$, and that only the voter knows the preimage x. The voter now submits both $\mathcal{E}_{pk}(x)$ and $\mathcal{E}_{pk}(\sigma_i)$ to $\mathfrak{BB}$. Before weeding, the hash is homomorphically evaluated on the ciphertext of the preimage, i.e., $\mathcal{E}_{pk}(h(x))$, followed by an equality test against the ciphertext of the credential $\mathcal{E}_{pk}(\sigma_i)$. A malicious authority can now cast only a valid ballot with a registered credential after the corresponding voter has cast a ballot, and an attempt to vote on their behalf is detectable in the weeding phase.

Acknowledgments

The authors acknowledge support from the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint project SURCVS. The project was also supported by the FNR INTER-VoteVerif and the FNR CORE project Q-CoDe.

References

1. Adida, B.: Helios: Web-based open-audit voting. In: USENIX security symposium. vol. 17, pp. 335–348 (2008)
2. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT 2015 - International Conference on the Theory and Applications of Cryptographic Techniques. pp. 430–454. Springer (2015)
3. Araújo, R., Barki, A., Brunet, S., Traoré, J.: Remote electronic voting can be efficient, verifiable and coercion-resistant. In: International Conference on Financial Cryptography and Data Security. pp. 224–232. Springer (2016)
4. Araújo, R., Foulle, S., Traoré, J.: A practical and secure coercion-resistant scheme for remote elections. In: Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2008)

5. Araújo, R., Rajeb, N.B., Robbana, R., Traoré, J., Youssfi, S.: Towards practical and secure coercion-resistant electronic elections. In: International Conference on Cryptology and Network Security. pp. 278–297. Springer (2010)
6. Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: Annual International Cryptology Conference. pp. 565–596. Springer (2018)
7. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory **6**(3), 13 (2014)
8. Carr, C., Costache, A., Davies, G.T., Gjøsteen, K., Strand, M.: Zero-knowledge proof of decryption for FHE ciphertexts. IACR Cryptology ePrint Archive **2018**, 26 (2018), `http://eprint.iacr.org/2018/026`
9. Catalano, D., Marcedone, A., Puglisi, O.: Authenticating computation on groups: New homomorphic primitives and applications. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 193–212. Springer (2014)
10. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: IEEE Symposium on Security and Privacy, 2008. pp. 354–368. IEEE (2008)
11. Cortier, V., Galindo, D., Küsters, R., Mueller, J., Truderung, T.: Sok: Verifiability notions for e-voting protocols. In: Security and Privacy (SP), 2016 IEEE Symposium on. pp. 779–798. IEEE (2016)
12. Delaune, S., Kremer, S., Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: Computer Security Foundations Workshop, 2006. 19th IEEE. pp. 12–pp. IEEE (2006)
13. Fiore, D., Gennaro, R., Pastro, V.: Efficiently verifiable computation on encrypted data. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 844–855. ACM (2014)
14. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: International Workshop on the Theory and Application of Cryptographic Techniques. pp. 244–251. Springer (1992)
15. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on Symposium on theory of computing-STOC\'09. pp. 169–169. ACM Press (2009)
16. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the aes circuit. In: Advances in Cryptology–CRYPTO 2012, pp. 850–867. Springer (2012)
17. Gjøsteen, K., Strand, M.: A roadmap to fully homomorphic elections: Stronger security, better verifiability. In: International Conference on Financial Cryptography and Data Security. pp. 404–418. Springer (2017)
18. Grontas, P., Pagourtzis, A., Zacharakis, A., Zhang, B.: Towards everlasting privacy and efficient coercion resistance in remote electronic voting. IACR Cryptology ePrint Archive **2018**, 215 (2018)
19. Jakobsson, M., Juels, A.: Mix and match: Secure function evaluation via ciphertexts. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 162–177. Springer (2000)
20. Jao, D., Soukharev, V.: Isogeny-based quantum-resistant undeniable signatures. In: International Workshop on Post-Quantum Cryptography. pp. 160–179. Springer (2014)
21. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society. pp. 61–70. ACM (2005)

22. Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion resistance and its applications 1. Journal of Computer Security **20**(6), 709–764 (2012)
23. Luo, F., Wang, K.: Verifiable decryption for fully homomorphic encryption. In: International Conference on Information Security. pp. 347–365. Springer (2018)
24. Roenne, P.B.: JCJ with improved verifiability guarantees. In: The International Conference on Electronic Voting E-Vote-ID 2016 (2016)
25. Smith, D.: New cryptographic voting schemes with best-known theoretical properties. In: Workshop on Frontiers in Electronic Elections (2005)
26. Spycher, O., Koenig, R., Haenni, R., Schläpfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In: International Conference on Financial Cryptography and Data Security. pp. 182–189. Springer (2011)
27. Strand, M.: A verifiable shuffle for the GSW cryptosystem. IACR Cryptology ePrint Archive **2018**, 27 (2018), `http://eprint.iacr.org/2018/027`
28. Sun, X., Tian, H., Wang, Y.: Toward quantum-resistant strong designated verifier signature from isogenies. In: Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference on. pp. 292–296. IEEE (2012)
29. Weber, S.G.: Coercion-Resistant Cryptographic Voting: Implementing Free and Secret Electronic Elections. VDM Publishing (2008)
30. Weber, S.G., Araujo, R., Buchmann, J.: On coercion-resistant electronic elections with linear work. In: Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. pp. 908–916. IEEE (2007)