Efficient Vote Authorization in Coercion-Resistant Internet Voting

Michael Schläpfer[1], Rolf Haenni[2], Reto Koenig[2,3], and Oliver Spycher[2,3]

[1] ETH Zurich, CH-8092 Zurich, Switzerland
michschl@inf.ethz.ch
[2] Bern University of Applied Sciences, CH-2501 Biel, Switzerland
{rolf.haenni,reto.koenig,oliver.spycher}@bfh.ch
[3] University of Fribourg, CH-1700 Fribourg, Switzerland
{reto.koenig,oliver.spycher}@unifr.ch

Abstract. Some years ago, Juels et al. introduced the first coercion-resistant Internet voting protocol. Its basic concept is still the most viable approach to address voter coercion and vote selling in Internet voting. However, one of the main open issues is its unrealistic computational requirements of the quadratic-time tallying procedure. In this paper, we examine the cause of this issue, namely the authorization of votes, and summarize the most recent proposals to perform this step in linear time. We explain the key underlying concepts of these proposals and introduce a new protocol based on anonymity sets. The size of these anonymity sets serves as an adjustable security parameter, which determines the degree of coercion-resistance. The main advantage of the new protocol is to move computational complexity introduced in recent works from the voter side to the tallying authority side.

1 Introduction

Remote voting over the Internet gains increasing attention as many governments aim at providing their citizens with electronic voting services. Although tremendous effort was put in research to understand the various aspects involved in electronic voting, no widely accepted solution to overcome all the security problems has been presented so far. One particular problem, namely voter coercion, was introduced in 2005 by Juels et al. in [13]. They propose a coercion-resistant Internet voting protocol to which we will simply refer as JCJ. Their protocol has been widely discussed and examined in the literature, and its basic concept still seems to be the most viable solution to address the voter coercion and vote selling problems. As appealing the approach may look in theory, it leaves some critical issues unanswered, for example the board-flooding problem or the quadratic running time of the tallying procedure. As we will not address the board-flooding problem in this paper, we refer to some of the most recent proposals in the literature [14].

This paper deals with the latter problem, i.e., the quadratic running time of the tallying procedure. We particularly focus on the two main building blocks of JCJ-based protocols, namely the elimination of *duplicate votes* and the detection of *fake votes*. These components are responsible for the expensive computations during tallying. Together with the elimination of invalid votes (those with invalid zero-knowledge proofs),

A. Kiayias and H. Lipmaa (Eds.): VoteID 2011, LNCS 7187, pp. 71–88, 2012.

we generally refer to these steps as *vote authorization*. We propose a modified protocol based on anonymity sets to address the efficiency problems of these building blocks and compare our protocol to some recent proposals for efficient vote authorization. In comparison with the closest recent protocol [6, 7], our approach is more expensive for the tallying authorities, but less expensive for voters. We consider this as an important property, because the computational resources on the voters' side are usually limited (e.g. in Internet voting, low-power devices or slow interpreted languages might be used by voters).

In Section 2, we first introduce the JCJ protocol, its critical building blocks, and recent improvements. In Section 3, we propose a modified protocol for efficient vote authorization. We compare the performance of our approach to existing protocols in Section 4. Finally, we summarize our conclusions and suggest some future work in Section 5.

2 Coercion-Resistant Internet Voting

Coercion-resistant Internet voting protocols follow in general the phases depicted in Figure 1. Prior to any voting event, eligible voters must register with the authority to get their credentials for participating at future elections. This phase is individual for the voters and usually carried out only once for a number of subsequent voting events. For every election, the following phases are repeatedly performed: *election setup*, *vote casting*, *vote authorization*, *tallying*. Note that some protocols allow to partly carry out vote authorization already during the vote casting phase.

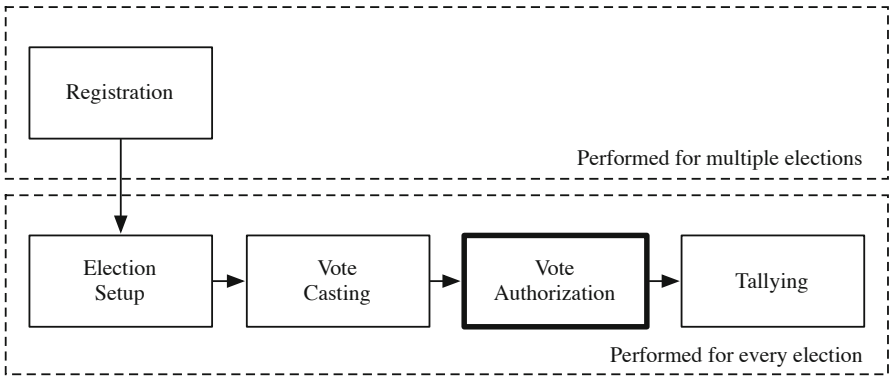

Fig. 1. Phases of coercion-resistant Internet voting protocols

While efficient and scalable solutions exist for most of the above phases, the vote authorization phase—as proposed by JCJ—requires computing capacities which grow quadratically in the number of votes cast. In a large-scale election setting such as nationwide parliamentary elections, this leads to unrealistic performance requirements. In the following, we explain the vote authorization phase in more detail, and then we provide a summary of the original JCJ protocol and of some recently proposed improvements.

2.1 Vote Authorization

Voting from a remote private place is inherently problematic with respect to privacy, because no privacy-preserving voting booth is used to protect it. Therefore, other measures must be taken to protect voters from coercion and to prevent vote selling. The main measures for this in JCJ are (a) the ability of coerced voters to create and cast fake votes, the coercer cannot distinguish from valid votes, and (b) the ability for voters to cast multiple votes, of which only one will be counted. Hence, various types of unauthorized votes may appear in the electronic ballot box and must thus be excluded before tallying. In some protocol descriptions, this phase is called *pre-tallying phase*, whereas sometimes it is implicitly included in the tallying phase. We call it *vote authorization phase*, which consists of the following three consecutive vote elimination steps:

Invalid Votes Elimination. The ballots created and cast by the voters usually include cryptographic zero-knowledge proofs for various purposes, for example for the correct construction of the encryptions, the correctness of the selected candidate choices, or the knowledge of the plaintexts. In the first step of the vote authorization phase, the authorities verify the validity of the proofs to exclude ballots with invalid proofs from further processing.

Duplicate Votes Elimination. In the second step, among all ballots that were cast with the same credential, exactly one is chosen for further processing according to some policy. In the "last-vote-counts" policy, for example, the last valid vote cast is selected to be included in the tallying, whereas all other votes from the same credential are excluded from further processing. This enforces the "one-voter-one-vote" principle.

Fake Votes Elimination. To conclude the vote authorization phase, all remaining fake votes have to be detected and removed. For this, the authorities need to check whether or not the ballots include credentials from eligible voters. Only the ballots passing this test are kept for the final tallying. Note that before eliminating fake votes, they must be unlinked from the actual votes cast. Otherwise, a coercer or vote buyer could easily detect the voter's attempt of not fulfilling the demands. The unlinking is usually achieved by shuffling the votes in a verifiable (re-encryption) mix-net.

2.2 The JCJ Protocol

In the following paragraphs, we briefly explain the phases of the JCJ protocol. Since we focus on the main building blocks, we settle for a semi-formal style of exposition. In particular, we do not thoroughly explain well-known cryptographic techniques. Furthermore, we assume the application of publicly verifiable group threshold mechanisms whenever registering or tallying authorities perform joint computations, even if the text might suggest a single entity. All ciphertexts are ElGamal encryptions over a pre-established multiplicative cyclic group $(\mathcal{G}_q, \cdot, 1)$ of order q, for which the decisional Diffie-Hellman problem (DDHP) is assumed to be hard. Note that the authors of the JCJ

protocol propose a modified version of ElGamal encryption for their formal proofs to work. The discussion in this paper is based on a simplified version with ordinary ElGamal encryptions as it is used by CIVITAS [9].

Registration. The *registrars* establish the random credential $\sigma \in \mathcal{G}_q$ and pass it to the voter via an *untappable channel*. Additionally, they append a public credential, i.e., a randomized encryption $S = \mathrm{Enc}_\varepsilon(\sigma, \alpha_S)$ of σ, to the voter's entry in the public voter roll, which is part of a public bulletin board. Value α_S denotes the encryption's randomness, and ε stands for the tallying authorities' common public key. Assuming a majority of trustworthy registrars, in the end only the voter will know σ and no one will know α_S.

Vote Casting. The voter identifies a choice c from the available set of valid choices (or candidates) $\mathcal{C}$. To cast the vote, the encryptions $A = \mathrm{Enc}_\varepsilon(\sigma, \alpha_A)$ and $B = \mathrm{Enc}_\varepsilon(c, \alpha_B)$ are posted to the public bulletin board, via an anonymous channel. The pair (A, B) must be accompanied by two non-interactive zero-knowledge proofs (NIZKP), one to prove knowledge of σ and one to prove $c \in \mathcal{C}$. Requiring the first proof prevents attackers from casting unauthorized votes by re-encrypting entries from the voter roll (recall that α_S is not known to anyone). Since each authorized vote on the voting board will be decrypted during the tallying phase, the second proof is needed to prevent coercers from forcing voters to select $c \notin \mathcal{C}$ according to some prescribed pattern, thus obtaining a receipt as described in [10].

To circumvent coercion, the voter can deceive the coercer by posting a fake vote to the voting board. To do so, the voter simply claims some arbitrary $\sigma' \in \mathcal{G}_q$ to be the proper credential and uses it to compute A. The encrypted vote B is computed according to the coercer's preference and the plaintexts of A and B are revealed to justify compliance. Alternatively, the voter can even let the coercer compute A and B and cast the vote using σ'.

Vote Authorization. At the end of the vote casting phase, the voting board contains a certain number N of votes cast, of which not all might make it to the final tally. First, the authorities verify all NIZKPs that were cast along with the votes. If a proof does not hold for a vote (A, B), it is marked accordingly and excluded from further processing (*invalid votes elimination*). Then the tallying authorities need to filter out votes that were cast multiple times with a proper credential (*duplicate votes elimination*) and votes that were cast with a fake credential (*fake votes elimination*). For both tasks, the authors of JCJ propose the application of *plaintext equivalence tests* (PET) [11]. Given two ElGamal encryptions $X = \mathrm{Enc}_\varepsilon(x, \alpha_X)$ and $Y = \mathrm{Enc}_\varepsilon(y, \alpha_Y)$, the group threshold algorithm $\mathrm{PET}(X, Y)$ returns *true* for $x = y$ and *false* for $x \neq y$, without revealing any information on x or y.[1]

Tallying. At the end, i.e., after eliminating all unauthorized votes, all remaining votes are jointly decrypted and counted. The final result is published.

[1] A common way of performing PET in a homomorphic encryption scheme is to check whether the decryption of $(X/Y)^z$ equals 1 for some random value z.

2.3 Improvements of the JCJ Protocol

Several proposals for improving the quadratic running-time of the tallying procedure in JCJ exist in the recent literature. In this subsection, we give a short overview of these developments. Two of them will be described in more details, as some of their ideas will re-appear in the description of our contribution in Section 3.

Smith, Weber [17, 19, 20]. Instead of applying $\mathrm{PET}(A_i, A_j)$, $1 \leq i, j \leq N$, on all pairs of distinct ciphertexts for removing duplicates, both Smith and Weber suggested computing and decrypting $A_1^z = \mathrm{Enc}_\varepsilon(\sigma_1^z), \ldots, A_N^z = \mathrm{Enc}_\varepsilon(\sigma_N^z)$, where z is a random value shared among the tallying authorities. The resulting *blinded* values σ_i^z are stored in a hash table for collision detection in linear time. Clearly, $\sigma_i = \sigma_j$ whenever $\sigma_i^z = \sigma_j^z$. In addition to eliminating duplicate votes, both authors propose using the same procedure for eliminating fake votes. In that case, however, based on the fact that the same exponent z is used across all ciphertexts, the coercer gets an attack strategy to identify whether a vote with known σ is counted [2, 9, 15]. Note that this attack does not apply to the elimination of duplicate votes.

Araujo et al. [1–4]. To solve the efficiency problem of the JCJ scheme, Araujo et al. suggest an approach based on group signatures. At registration, voters obtain their credentials. Unlike JCJ, no public values are related to voter roll entries. Their credentials enable the voters to deduce invalid credentials and mislead coercers. If the provided proofs hold, duplicates on the voting board are publicly identifiable by the equality of two values that are cast along with the vote. After mixing the relevant values on the voting board, the tallying authorities use their private keys to identify the legitimate votes. Notably, all information on their legitimacy is sent along with the vote itself, but can only be assessed by a sufficiently large group of tallying authorities. Fully avoiding plaintext equivalence tests between cast values and voter roll entries summarizes the essence of this elegant approach to avoid the inefficient comparison procedure.

An inherent weakness of this approach is the fact that a majority of colluding registrars could compute valid (but illegitimate) credentials unnoticed. As described earlier, adding illegitimate votes to the tally in JCJ requires the knowledge of a credential σ that complies with an entry S in the voter roll, i.e., such attacks could easily be detected. This is not the case in Araujo et al.'s scheme. Nevertheless, we believe that the approach holds much potential.

Spycher et al. [18]. This protocol strongly relates to the original JCJ protocol. For removing duplicates, they suggest using the linear-time scheme proposed by Smith and Weber. For fake vote elimination, they suggest preserving the use of the voter roll. The key to better efficiency lies in requiring voters to indicate which voter roll entry their vote (A, B) relates to. Tallying authorities then apply PET only on respective re-encryptions of A and S, where S is the public credential copied from the indicated voter roll entry. To preserve privacy, a certain number of fake votes is introduced by the authorities for each voter roll entry. This allows voters to deny the fact of having submitted their vote. Vote authorization thus becomes linear over the total number of submitted votes. More details on each steps of this protocol are given below.

Registration. The registration step is conducted according to JCJ. Additionally, it is assumed that a distinct public number, for example the index $i \in \{1, \ldots, n\}$ of the voter's entry in the voter roll, is assigned to each voter.

Vote Casting. To cast a vote, the voters perform the same steps as in JCJ. In addition to the values A and B along with corresponding proofs, the value $C = \mathrm{Enc}_\varepsilon(i, \alpha_C)$, accompanied by a NIZKP to prove knowledge of i, is posted to the bulletin board. The authorities later use i to locate the public credential S on the voter roll and thus to perform a single PET to efficiently eliminate fake votes. Note that the voting board must also accept encryptions C of values different from the voter's public number i.

Vote Authorization. After excluding votes with invalid proofs, the authorities add a random number β_i of additional fake votes for each voter (let β denote the average number of additional votes per voter). After duplicate elimination by applying Smith's and Weber's scheme on values A, the resulting adjusted list is passed as input to a first re-encryption mix-net, which outputs tuples (A', B', C'). Next, the authorities decrypt C' to extract i and establish a list of tuples (A', B', S). Votes for which the decryption renders an invalid index $i \notin \{1, \ldots, n\}$ are excluded from further processing. The remaining tuples are passed to a second re-encryption mix-net, which outputs tuples (A'', B'', S'). Now the tallying authorities perform $\mathrm{PET}(A'', S')$ for each tuple. If the algorithm returns *false*, which is the case for all fake votes, the tuple is marked to be excluded from further processing.

Tallying. At the end, the tallying authorities jointly decrypt the values B'' of all remaining votes and publish the result.

Selections [6, 7]. Although SELECTIONS is based on JCJ, the approach presented by Clark and Hengartner has a slightly different setting. We will shortly summarize its phases and point out the differences.

Registration. The public credential is not an encryption of the voter's credential σ, but an encryption of g^σ for a publicly known generator g, i.e., $S = \mathrm{Enc}_\varepsilon(g^\sigma, \alpha_S)$.

Election Setup. For every election, the public credentials are re-randomized into $S' = \mathrm{Enc}_\varepsilon(\hat{g}^\sigma, \alpha_{S'})$, where $\hat{g} = g^\alpha$ is a fresh generator and $\alpha_{S'} = \alpha_S \cdot \alpha$ the new randomization for a random (but unknown) value α. The fresh generator $\hat{g}$ is also used for casting votes. This mechanism prevents information leakage across elections.

Vote Casting. When casting a vote, the voter sends a commitment $A = \hat{g}^\sigma$, the encrypted vote $B = \mathrm{Enc}_\varepsilon(c, \alpha_B)$, and a re-encryption of the public credential $C = \mathrm{ReEnc}_\varepsilon(S', \alpha_C)$ to the public bulletin board. Additionally, an *anonymity set* containing S' and $\beta - 1$ randomly chosen public credentials different from S' is selected. Then the voter constructs a NIZKP to prove that C is a re-encryption of one of the β public credentials in the anonymity set. This proof together with a proof of knowledge of σ and a proof of well-formedness for $c \in \mathcal{C}$ are added to the ballot.

Vote Authorization. Similarly to JCJ, the tallying authorities first verify the proofs related to the submitted votes. If a proof does not hold for a vote, it is marked and excluded from further processing, i.e., invalid votes are eliminated. This can be performed during the vote casting phase, in particular at the moment the individual ballots arrive on the public bulletin board. The detection and elimination of duplicate votes follows from the simple fact that votes with the same credential will have the same commitment $A = \hat{g}^{\sigma}$. In that case, only one vote is kept for further processing (according to some policy). After eliminating duplicate votes, the remaining tuples (A, B, C) are mixed and re-encrypted (the commitment A is treated as an encryption with randomness 0) into (A', B', C'). For fake vote elimination, a simple PET between A' and C' is performed. If $\mathrm{PET}(A', C')$ returns false, which is the case for all fake votes, the vote is marked and excluded from further processing.

Tallying. At the end, all remaining votes B' are jointly decrypted and tallied. The final result is published.

3 A New Protocol Based on Anonymity Sets

Similar to the protocols of Spycher et al. and SELECTIONS, our protocol strongly relates to the original JCJ protocol. For eliminating duplicate votes, we propose using the linear-time scheme of Smith and Weber, and for fake votes elimination, we suggest linking the vote to the voter roll. Instead of explicitly introducing fake votes by the authorities, we require the voter to specify an *anonymity set* of voters, similar to the one in SELECTIONS. The difference is that in our protocol, the voter only claims to be one of β voters without any proof for this claim. In contrast to SELECTIONS, the voter is thus not required to construct an expensive zero-knowledge proof during vote casting. In the vote authorization phase, the authorities replicate every submitted ballot into β ballots, one for each voter in the anonymity set. In other words, every submitted proper ballot leads to one authorized vote and $\beta - 1$ implicit fake votes. The protocol can therefore be regarded as a synthesis between SELECTIONS and the protocol of Spycher et al., where β, the size of the anonymity sets, constitutes an adjustable security parameter that determines the degree of coercion-resistance. More details on this idea are given in the subsequent description of the protocol. An overview of the protocol is depicted in Figure 2.

3.1 Protocol Description

In our description of the protocol, we follow the same style of exposition and notation as for the protocols described in Section 2. As many elements strongly relate to the original JCJ protocol and its successors, we will not explain everything again in comprehensive detail. Note that our protocol, in contrast with SELECTIONS, does not require a particular election setup phase.

Registration. The registration step is conducted according to Spycher et al. It is thus assumed that a distinct public number, for example the index $i \in \{1, \ldots, n\}$ of the voter's entry in the voter roll, is assigned to each voter.

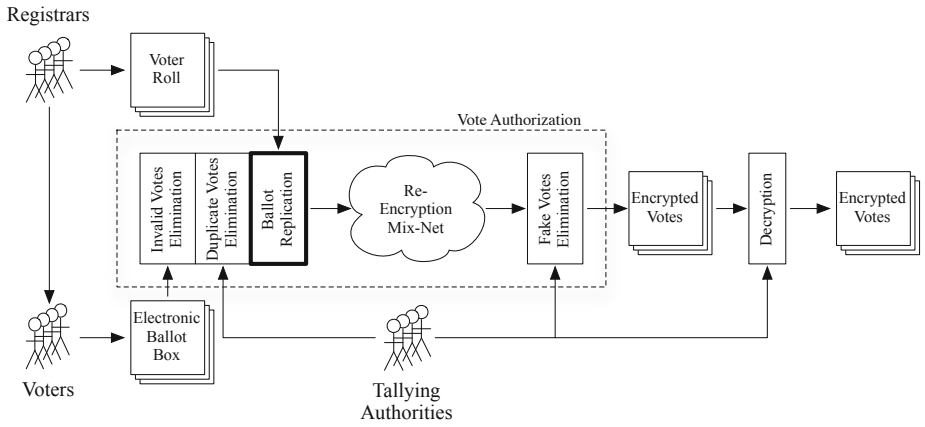

Fig. 2. Overview of the new protocol with some details about the vote authorization phase

Vote Casting. To cast a vote, the voter performs the same steps as in JCJ. Additionally to posting values A and B along with corresponding proofs, a subset $I \subseteq \{1, \ldots, n\}$ of size β is chosen at random and added to the ballot. This is the ballot's anonymity set, which must include the voter's own index i to become a vote that counts. Since the voting board is a public bulletin board, the voter is able to individually verify that the vote has been cast as intended and recorded as cast.

Vote Authorization. In a first step, all votes with invalid proofs are marked to be excluded from further processing. Then duplicate votes are eliminated by applying Smith's and Weber's scheme on values A. Next, the authorities create β new ballots (A, B, S_j) for every submitted ballot and for every $j \in I$ in the corresponding anonymity set. These ballots are published on the voting board and thus, the correct construction is universally verifiable. The adjusted list of ballots is passed as input to a verifiable re-encryption mix-net, which outputs tuples (A', B', S'_j). Now the authorities perform $\mathrm{PET}(A', S'_j)$ for each tuple. All ballots for which the algorithm returns $false$ are marked and excluded from further processing.

Tallying. The remaining unmarked ballots are authorized as legitimate votes and for every such ballot, B' is jointly decrypted and counted.

3.2 Security

Our protocol strongly relates to SELECTIONS and generally the same security considerations apply to our protocol. We do neither argue about the registration phase that corresponds to the original JCJ protocol nor about the final talling phase, which essentially consists of decrypting the authorized ballots and counting them.[2] In the following,

[2] The security properties of the tallying phase depend on the encryption scheme that is used and is not related to the security properties of our protocol (as long as the encryption scheme allows verifiable mixing and does not allow the coercer to link the cast vote to the mixed ballots).

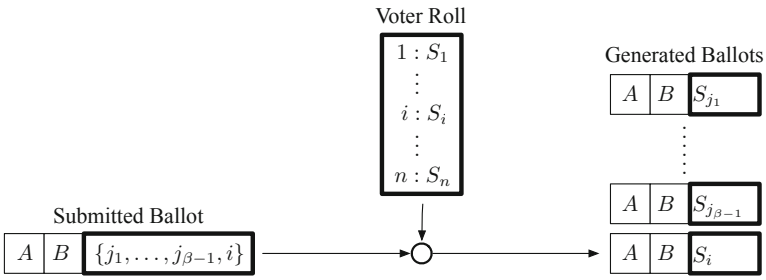

Fig. 3. The ballot replication step for one submitted ballot

we will briefly explain how vote casting and vote authorization in our protocol relates to SELECTIONS and recap the security considerations of SELECTIONS with respect to the security parameter β.

Vote Casting. As it is the case for every coercion-resistant voting protocol, the voter must have one moment of privacy to cast the real vote. Hence, the coercer can only observe the public bulletin board to check for the voter's compliance. During the vote casting phase, the public bulletin board reveals the same information to a coercer as it is the case for SELECTIONS. Particularly, the coercer knows the number of votes associated with each voter. The greater the security parameter β, the more likely the coerced voter's index occurs in someone else's anonymity set and thus the less certain the coercer will be regarding the voter's compliance.

Vote Authorization. The coercer does not get additional information on whether the vote will be tallied or not from observing the elimination of invalid votes. The next step, the elimination of duplicate votes using the technique of Smith and Weber is the same as in the scheme of Spycher et al. and is similar to SELECTIONS, i.e., the same security considerations apply. The ballot replication step is different to all existing approaches. However, the straightforward inflation of the submitted ballots with the indices of the anonymity set is public and does obviously not reveal any additional information to the coercer. The last step of the vote authorization phase, the mixing of the remaining ballots and the elimination of fake votes, is again the same as in Spycher et al. and SELECTIONS, i.e., the same security arguments apply here. In particular, because of the fact that the coercer cannot link the eliminated ballots to the initial votes, no information is leaked to the coercer on whether the coerced voter's fake vote was finally accepted or not.

Security Parameter β. Clark and Hengartner provide a thorough analysis regarding the security parameter β (size of the anonymity sets) in SELECTIONS [7]. The exact same considerations can be applied for our approach. There are essentially three interesting cases:

1. The anonymity set includes the entire voter roll, that is $\beta = n$. In this case, the degree of coercion-resistance corresponds to JCJ, but vote authorization (or more precisely fake vote elimination) is again quadratic in n (or more precisely multilinear in n and N, see Table 3).
2. The anonymity set has a fixed size, for example $\beta = 50$. Clark and Hengartner showed that a coercer may decide with small but non-negligible probability whether or not the coerced voter complied with the instructions. Hence, if β is a constant value, coercion-resistance is not given to the full, but depending on the value of β, to a reasonable extent.
3. The size of the anonymity set varies among voters, but has a minimal size $\beta_i \geq \beta$, for example $\beta_i \geq 50$. Clark and Hengartner point out the possibility of coerced voters to place their real vote as *stealth votes* with $\beta_i = n$. They also emphasize that in this situation, they need to assume that other stealth votes are cast as well to assure *adversarial uncertainty*.

Temporal Aspects. The security experiments of JCJ do not fully capture some important temporal aspects. For example, it is assumed that all honest voters submit their ballots in only one step (i.e., in parallel). In a more realistic setting, the coercer may observe the order and time when the ballots arrive on the public bulletin board. Consider for example the case where the coerced voter has only a short moment of privacy during night time, when only few other voters are casting votes. Observing the public bulletin board during this time, the coercer might get a strong indication whether or not the coerced voter really complied. The problem is especially problematical in both SELECTIONS and our approach, since the probability that the coerced voter appears in another voter's anonymity set might be low (depending on the actual choice of the security parameter β). To counteract the corresponding advantage for the coercer, we propose the following extension to our approach:

1. Before casting the ballot, the voter encrypts the anonymity set with the public key of the tallying authorities.
2. After eliminating duplicates, but before ballot replication, the ballots are mixed and re-encrypted in an additional re-encryption mix-net (similar to Spycher et al.).
3. The tallying authorities jointly decrypt the anonymity sets included in the mixed ballots.

Formal Proofs. Coercion-resistance of our protocol can be proved under the game-based definition of Juels et al. [13]. Since our approach is close to the approach presented by Clark and Hengartner, their security games serve as a starting point for the formal proofs, which we will carry out as future work.

4 Performance Comparison

This section is dedicated to a performance comparison of all the above introduced approaches. Our comparison excludes the one-time registration phase. In our results, the number of registered voters is denoted by n, the number of submitted ballot by N, the number of mixing and tallying authorities by T, the number of candidates by m, and

the size of the anonymity sets by β. We take the work of Clark and Hengartner [6, 7] as a starting point and augment their findings with the performance properties of our protocol and the one of Spycher et al. We make similar assumptions to facilitate a better comparison:

- We only use standard ElGamal encryption over a modular multiplicative group of integers (i.e., not the modified ElGamal version of JCJ).
- We only count the number of necessary modular exponentiations to perform the respective tasks (i.e., all other arithmetic operations are neglected).
- We do not use techniques, which could equally improve the performance of all protocols (e.g., the "blocking technique" of CIVITAS).
- We assume that a valid vote consists of exactly one candidate $c \in \mathcal{C}$, where $m = |\mathcal{C}|$ denotes number of candidates.
- We assume that the re-encryption mix-nets use *randomized partial checking* [12] for proving the correctness of the mixing (i.e., each authority mixes the encryptions twice and half of these re-encryptions are checked).
- We assume that all encrypted votes are decrypted during the tallying phase (i.e., no homomorphic tallying).
- We assume that all tallying authorities participate at the vote authorization phase (e.g., distributed instead of threshold decryption or PET).
- We assume that all commitments in the distributed operations are based on hash functions.

In contrast to Clark and Hengartner, we include the proofs of well-formedness of the encrypted votes in our calculations. We also take the election setup of their protocol and the tallying phase into consideration. For improving the readability of the results, we have re-arranged the values for checking the proofs in a separate *verification phase*. To simplify the results given in [6, 7], we consider only the worst case when all submitted ballots reach the fake vote elimination phase. In other words, we assume that all submitted ballots contain valid proofs and that the ballot box contains no duplicate votes (but searching for invalid and duplicate votes is still necessary). We also assume that every registered voter has submitted at least one valid vote, i.e., the number of votes to decrypt during tallying is exactly n and $N \geq n$.

4.1 Performance Analysis

Table 2 and Table 3 summarize the results of the performance analysis. Table 2 is based on corresponding values for the cryptographic primitives as given in Table 1, and Table 3 shows the same results more compactly in Big-Oh notation. Note that some of the values in Table 1 slightly differ from the ones given in [6, 7]. Since proving knowledge of a plaintext requires only a proof of knowledge of the encryption randomness (Schnorr), it can be constructed with 1 and verified with 2 exponentiations. Proving the correct decryption corresponds to proving a single equality of discrete logarithms (Chaum-Pederson). In the distributed case with T authorities, this simply scales up to T, $2T$, and $4T$ exponentiations for performing the partial decryptions, constructing the proofs, and verifying the proofs, respectively. Finally, a proof of encrypting 1-out-of-m plaintexts requires $4m-2$ exponentiations to construct (2 for the correct value and 4 for

Table 1. Number of necessary exponentiations for various cryptographic primitives according to the procedures as described in [5, 8, 16]

	Perform Operation	Generate Proof(s)	Verify Proof(s)
Encryption: Standard ElGamal encryption with proof of knowledge of plaintext (Schnorr).	2	1	2
Well-Formed Encryption: Standard ElGamal encryption of 1-out-of-m possible plaintexts with proof of well-formedness (Chaum-Pederson, OR-composition).	2	$4m-2$	$4m$
Re-Encryption: Standard ElGamal re-encryption with proof of correctness (Chaum-Pederson).	2	2	4
Well-Formed Re-Encryption: Standard ElGamal re-encryption of 1-out-of-β possible encryptions with proof of well-formedness (Chaum-Pederson).	2	$4\beta-2$	4β
Mixing: Re-encryption and permutation of N ciphertext tuples of length k.	$2kN$	$2kN$	$4kN$
Mix-Net: T authorities perform a re-encryption mix-net with randomized partial checking (double re-encryption, but only half of the proofs are provided).	$4kNT$	$2kNT$	$4kNT$
Commitment: Applying an exponent with proof of knowledge (Schnorr).	1	1	2
Distributed Commitment: T authorities applying exponents with proofs of knowledge (Schnorr).	T	T	$2T$
Blinding: Applying an exponent on the ciphertext with proof of correctness (Chaum-Pederson).	2	2	4
Distributed Blinding: T trustees applying exponents on the ciphertexts with proofs of correctness (Chaum-Pederson).	$2T$	$2T$	$4T$
Decryption: Standard ElGamal decryption with proof of correctness (Chaum-Pederson).	1	2	4
Distributed Decryption: T trustees performing distributed ElGamal decryptions with proofs of correctness (Chaum-Pederson).	T	$2T$	$4T$
Plaintext Equivalence Test: Distributed blinding followed by distributed decryption.	$3T$	$4T$	$8T$

every $m-1$ simulated values in the OR-composition) and $4m$ exponentiations to verify. Similarly, proving the re-encryption of 1-out-of-β ciphertexts requires $4\beta-2$ exponentiations to construct and 4β to verify the proof. The results given in Table 2 are based on these modifications, but they have no impact on the asymptotic results of Table 3. Note that the performance of the final tallying by decrypting the valid votes in a distributed way is the same in all protocols (nT exponentiations for performing the decryptions, $2nT$ for constructing the proofs, and $4nT$ for verifying the proofs). More details about the performance calculations are given in the upcoming paragraphs.

*a) JCJ (*CIVITAS*).* Vote casting consist of two encryptions ($= 4$) with one proof of knowledge of plaintext ($= 1$) and one proof of well-formedness ($= 4m-2$). Vote authorization requires verifying N proofs of knowledge ($= 2N$) and N proofs of well-formedness ($= 4mN$) to eliminate invalid votes, $\binom{N}{2}$ many PETs ($= \frac{3}{2}(N^2-N)T$)

Table 2. Performance comparison by counting the number of modular exponentiations required in each phase

	JCJ (CIVITAS)	Araujo et al.	Spycher et al.	Clark et al. (SELECTIONS)	Our Protocol
Election Setup	–	–	–	$(4n+2)T$	–
Vote Casting	$4m+3$	$4m+13$	$4m+6$	$4\beta+4m+2$	$4m+3$
Vote Authorization					
Eliminate Invalid Votes	$(4m+2)N$	$(4m+10)N$	$(4m+4)N$	$(4\beta+4m+2)N$	$(4m+2)N$
Insert Fake Votes	–	–	$6\beta n$	–	–
Elim. Duplicate Votes	$\frac{7}{2}(N^2-N)T$	0	$7(N+\beta n)T$	0	$7NT$
1st Mixing of Ballots	$12NT$	$30NT$	$18(N+\beta n)T$	$18NT$	$18\beta NT$
2nd Mixing of Ballots	–	–	$21(N+\beta n)T$	–	–
Mixing of Credentials	$6NT$	–	–	–	–
Eliminate Fake Votes	$7nNT$	$(14T+6)N$	$7(N+\beta n)T$	$7NT$	$7\beta NT$
Tallying	$3nT$	$3nT$	$3nT$	$3nT$	$3nT$
Verification					
Election Setup	–	–	–	$4(n+1)T$	–
Eliminate Invalid Votes	$(4m+2)N$	$(4m+10)N$	$(4m+4)N$	$(4\beta+4m+2)N$	$(4m+2)N$
Elim. Duplicate Votes	$4(N^2-N)T$	0	$8(N+\beta n)T$	0	$8NT$
1st Mixing of Ballots	$8NT$	$20NT$	$12(N+\beta n)T$	$12NT$	$12\beta NT$
2nd Mixing of Ballots	–	–	$16(N+\beta n)T$	–	–
Mixing of Credentials	$4NT$	–	–	–	–
Eliminate Fake Votes	$8nNT$	$(16T+8)N$	$8(N+\beta n)T$	$8NT$	$8\beta NT$
Tallying	$4nT$	$4nT$	$4nT$	$4nT$	$4nT$

with proofs ($= 2(N^2-N)T$) to eliminate duplicates, a re-encryption mix-net for N encryption pairs ($= 8NT$) with proofs ($= 4NT$) to mix the ballots, a second re-encryption mix-net for n single encryptions ($= 4NT$) with proofs ($= 2NT$) to mix the credentials, and finally nN many PETs ($= 3nNT$) with proofs ($= 4nNT$) to eliminate fake votes. Corresponding values for verifying the proofs follow accordingly.

b) Araujo et al. We use the latest version of the protocol for the comparison [3] and adopt the analysis provided in [7]. Vote casting consists of four encryptions ($= 8$) with three proofs of knowledge of plaintext ($= 3$) and one proof of well-formedness ($= 4m - 2$).

Table 3. Performance comparison by describing the asymptotic growth of the numbers of exponentiations in each phase using the Big-Oh notation (relative to n, N, β, m, and T)

	JCJ (CIVITAS)	Araujo et al.	Spycher et al.	Clark et al. (SELECTIONS)	Our Protocol
Election Setup	–	–	–	nT	–
Vote Casting	m	m	m	$\beta{+}m$	m
Vote Authorization	$N^2T{+}nNT{+}mN$	$NT{+}mN$	$NT{+}\beta nT{+}mN$	$NT{+}\beta N{+}mN$	$\beta NT{+}mN$
$T, m =$ const.	$N^2{+}nN$	N	$N{+}\beta n$	βN	βN
Tallying	nT	nT	nT	nT	nT
Verification	$N^2T{+}nNT{+}mN$	$NT{+}nT{+}mN$	$NT{+}\beta nT{+}mN$	$NT{+}\beta N{+}nT{+}mN$	$\beta NT{+}nT{+}mN$
$T, m =$ const.	$N^2{+}nN$	$N{+}n$	$N{+}\beta n$	$\beta N + n$	$\beta N + n$

Two of the encrypted values and one of the non-encrypted values need one exponentiation to compute ($= 3$). A proof of representation that relates the non-encrypted to one of the encrypted values ($= 1$) is added to the ballot. To eliminate invalid votes, vote authorization requires verifying $3N$ proofs of knowledge ($= 6N$), N proofs of well-formedness ($= 4mN$), and N proofs of representation ($= 4N$). Duplicates can be removed at no additional costs. As suggested in [7], we omit the additional encryption step, which can be performed by the first mix-net authority. The re-encryption mix-net takes N encryption 5-tuples ($= 30NT$) as input and produces corresponding proofs ($= 20NT$). Finally, eliminating fake votes requires for each vote two commitments ($= 2N$), two Chaum-Pederson proofs ($= 4N$), and two PETs ($= 6NT$) with proofs ($= 8NT$). Corresponding values for verifying the proofs follow accordingly.

c) Spycher et al. Vote casting consist of three encryptions ($= 6$) with two proofs of knowledge of plaintext ($= 2$) and one proof of well-formedness ($= 4m{-}2$). Vote authorization requires verifying $2N$ proofs of knowledge ($= 4N$) and N proofs of well-formedness ($= 4mN$) to eliminate invalid votes, three encryptions without proofs for each of the βn inserted fake votes ($= 6\beta n$), $N{+}\beta n$ many distributed blinding operations ($= 2(N{+}\beta n)T$) with proofs ($= 2(N{+}\beta n)T$) and distributed decryptions ($= (N{+}\beta n)T$) with proofs ($= 2(N{+}\beta n)T$) to eliminate duplicates (Smith's and Weber's scheme), a re-encryption mix-net for $N{+}\beta n$ encryption triples ($= 12(N{+}\beta n)T$) with proofs ($= 6(N{+}\beta n)T$) to mix the ballots, $N{+}\beta n$ distributed decryptions ($= (N{+}\beta n)T$) with proofs $2(N{+}\beta n)T$ plus another re-encryption mix-net for $N{+}\beta n$ encryption triples ($= 12(N{+}\beta n)T$) with proofs ($= 6(N{+}\beta n)T$) to mix the ballots a second time, and finally $N{+}\beta n$ many PETs ($= 3(N{+}\beta n)T$) with proofs ($= 4(N{+}\beta n)T$) to eliminate fake votes. Corresponding values for verifying the proofs follow accordingly.

*d) Clark et al. (*SELECTIONS*).* The election setup requires n distributed blinding operations ($= 2nT$), one distributed commitment ($= T$) and an AND-composition of corresponding proofs ($= 2nT + T$). Vote casting consist of a commitment ($= 1$) with a proof of knowledge ($= 1$), a re-encryption ($= 2$) with a proof of well-formedness ($= 4\beta - 2$), and one encryption ($= 2$) with a proof of well-formedness ($= 4m-2$). Vote authorization requires a re-encryption mix-net for N encryption triples ($= 12NT$) with proofs ($= 6NT$) to mix the ballots, and finally N many PETs ($= 3NT$) with proofs ($= 4NT$) to eliminate fake votes. Corresponding values for verifying the proofs follow accordingly.

e) Our Protocol. Vote casting consist of two encryptions ($= 4$) with one proof of knowledge of plaintext ($= 1$) and one proof of well-formedness ($= 4m-2$). Vote authorization requires verifying N proofs of knowledge ($= 2N$) and N proofs of well-formedness ($= 4mN$) to eliminate invalid votes, N many distributed blinding operations ($= 2NT$) with proofs ($= 2NT$) and distributed decryptions ($= NT$) with proofs ($= 2NT$) to eliminate duplicates (Smith's and Weber's scheme), a re-encryption mix-net for βN encryption triples ($= 12\beta NT$) with proofs ($= 6\beta NT$) to mix the ballots, and finally βN many PETs ($= 3\beta NT$) with proofs ($= 4\beta NT$). Corresponding values for verifying the proofs follow accordingly.

4.2 Discussion

In our new protocol, casting a vote is as efficient as in the original JCJ protocol or in CIVITAS. For a fixed candidate set, a constant number of exponentiations is needed. If the candidate set is reasonably small, this seems to be feasible on today's typical client platforms (e.g., 11 exponentiations are needed for $m = 2$ choices in a referendum). Compared to SELECTIONS, where casting a vote depends on the security parameter β, this is the main advantage of our approach. We think that for reasonably large anonymity sets, the protocol of Clark et al. is not competitive enough to be considered as a solution for a coercion-resistant voting system (e.g., 210 exponentiations are needed for $\beta = 50$ and $m = 2$). Asymptotically, the number of exponentiations is $O(\beta+m)$ for SELECTIONS and $O(m)$ for all the others (see Table 3).

To determine the protocol with the most efficient vote authorization procedure, for example by interpreting the general asymptotic results in Table 3, we need to take into account multiple systems parameters. To facilitate this task, we propose two simplifications: we consider a constant number of authorities and a fixed candidate set (both T and m affect all protocols in a similar way). The corresponding simplified growth rates are shown in Table 3 below the general results. While JCJ and CIVITAS are essentially quadratic in N, it turns out that Araujo et al.'s protocol—although it has relatively high constant factors—is the only one that is truly linear in N. Among the others, the advantage of Spycher et al's protocol is the fact that β only multiplies with n, the number of voters, which is fixed for a given election (whereas the number of submitted ballots N has no upper bound). On the other hand, Spycher et al's protocol has the least favorable constant factors among all. Our new protocol and SELECTIONS are comparable with respect to their growth rates, but SELECTIONS has a significantly lower constant

factor for βN. However, SELECTIONS allows to carry out the invalid votes elimination already during the vote casting phase, which is a considerable advantage compared to our protocol. Note that the same conclusions hold for the verification procedure.

5 Conclusion

We have presented a new improvement of the JCJ protocol that allows efficient vote authorization without requiring more computation power on the voter's side. Conceptually, it is a mix between the existing protocols of Spycher et al. and SELECTIONS. To conclude this paper, we summarize the phases with emphasis on vote authorization and compare our protocol with the different approaches discussed in this paper.

Election Setup. In contrast to SELECTIONS, our protocol as well as the other examined protocols require no election setup.

Vote Casting. Compared with the protocol of Spycher et al. we do not require the authorities to generate random fake votes during the vote casting phase and therefore we reduce the effort for the authorities in this phase. In contrast to SELECTIONS, our improvement introduces no additional effort for the voter in terms of modular exponentiations. The voter only has to add a set of β voter roll indices to the ballot. When applied on systems and technologies with limited computing resources such as mobile phones or web applications using JavaScript, all additional performance requirements are undesirable. Another advantage of our approach is the fact, that security parameter β does not affect the client-side at all. We believe that security should not be bounded by the computing equipment of the individual voters or even require them to buy better computers to protect their privacy or an e-voting protocol to a reasonable degree. Moreover, this contradicts the fundamental principle of equality. In our approach β only affects the server-side performance requirements which is more scalable with respect to computation power.

Vote Authorization. The lower computation requirements for the voters during the vote casting phase yield more effort to put in the vote authorization phase. Security parameter β affects the computational requirements on the server-side as a linear factor. In particular, we need to explicitly remove duplicates using the linear approach of Smith and Weber and we enlarge the input of the mix-net by factor β. Hence, mixing in our protocol requires additional computing power compared to the other protocols.

Current and Future Work. Since our protocol strongly relates to existing proven concepts, we informally justified the correctness of the individual phases. However, future work includes formal proofs of correctness of these arguments. Currently, we are engaged in developing prototypes for various coercion-resistant voting protocols. Our experience with these realizations will allow us to compare the existing approaches from a more practical perspective.

Acknowledgments. We thank Jeremy Clark and the three anonymous reviewers for their constructive comments. This research is supported by the *Swiss Federal Chancellery*, the *Hasler Foundation* (project No. 09037), and the *Mittelbauförderung* of the Bern University of Applied Sciences.

References

1. Araujo, R.: On Remote and Voter-Verifiable Voting. PhD thesis, Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany (2008)
2. Araújo, R., Foulle, S., Traoré, J.: A practical and secure coercion-resistant scheme for remote elections. In: Chaum, D., Kutylowski, M., Rivest, R.L., Ryan, P.Y.A. (eds.) FEE 2007, Frontiers of Electronic Voting, Schloss Dagstuhl, Germany, pp. 330–342 (2007)
3. Araújo, R., Foulle, S., Traoré, J.: A Practical and Secure Coercion-Resistant Scheme for Internet Voting. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections. LNCS, vol. 6000, pp. 330–342. Springer, Heidelberg (2010)
4. 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)
5. Brandt, F.: Efficient Cryptographic Protocol Design Based on Distributed El Gamal Encryption. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 32–47. Springer, Heidelberg (2006)
6. Clark, J.: Democracy Enhancing Technologies: Toward Deployable and Incoercible E2E Elections. PhD thesis, University of Waterloo, Canada (2011)
7. Clark, J., Hengartner, U.: Selections: Internet Voting with Over-the-Shoulder Coercion-Resistance. In: Danezis, G. (ed.) FC 2011. LNCS, vol. 7035, pp. 47–61. Springer, Heidelberg (2012)
8. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. Technical Report TR 2007-2081, Department of Computer Science. Cornell University (2007)
9. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: SP 2008, 29th IEEE Symposium on Security and Privacy, Oakland, USA, pp. 354–368 (2008)
10. Di Cosmo, R.: On privacy and anonymity in electronic and non electronic voting: the ballot-as-signature attack. Hyper Articles en Ligne, hal-00142440(2) (2007)
11. 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)
12. Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In: Boneh, D. (ed.) SS 2002, 11th USENIX Security Symposium, San Francisco, USA, pp. 339–353 (2002)
13. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Atluri, V., De Capitani di Vimercati, S., Dingledine, R. (eds.) WPES 2005, 4th ACM Workshop on Privacy in the Electronic Society, Alexandria, USA, pp. 61–70 (2005)
14. Koenig, R., Haenni, R., Fischli, S.: Preventing Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes. In: Camenisch, J., Fischer-Hübner, S., Murayama, Y., Portmann, A., Rieder, C. (eds.) SEC 2011. IFIP AICT, vol. 354, pp. 116–127. Springer, Heidelberg (2011)
15. Pfitzmann, B.: Breaking an Efficient Anonymous Channel. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 332–340. Springer, Heidelberg (1995)

16. Rjasková, Z.: Electronic voting schemes. Diploma thesis, Department of Computer Science. Comenius University, Bratislava, Slovak Republic (2002)
17. Smith, W.D.: New cryptographic voting scheme with best-known theoretical properties. In: FEE 2005, Workshop on Frontiers in Electronic Elections, Milan, Italy (2005)
18. Spycher, O., Koenig, R., Haenni, R., Schläpfer, M.: A New Approach towards Coercion-Resistant Remote E-Voting in Linear Time. In: Danezis, G. (ed.) FC 2011. LNCS, vol. 7035, pp. 182–189. Springer, Heidelberg (2012)
19. Weber, G., Araujo, R., Buchmann, J.: On coercion-resistant electronic elections with linear work. In: ARES 2007, 2nd International Conference on Availability, Reliability and Security, Vienna, Austria, pp. 908–916 (2007)
20. Weber, S.: Coercion-Resistant Cryptographic Voting: Implementing Free and Secret Electronic Elections. VDM Verlag, Saarbrücken (2008)