Efficiency Evaluation of Cryptographic Protocols for Boardroom Voting

Oksana Kulyk*, Stephan Neumann*, Jurlind Budurushi*, Melanie Volkamer*‡,
Rolf Haenni†, Reto Koenig†, Philemon von Bergen†
*Technische Universität Darmstadt/CASED, Darmstadt, Germany
Email: name.surname@secuso.org
†Bern University of Applied Sciences, Bern, Switzerland
Email: name.surname@bfh.ch
‡Karlstad University, Karlstad, Sweden

***Abstract*—Efficiency is the bottleneck of many cryptographic protocols towards their practical application in different contexts. This holds true also in the context of electronic voting, where cryptographic protocols are used to ensure a diversity of security requirements, e.g. secrecy and integrity of cast votes. A new and promising application area of electronic voting is boardroom voting, which in practice takes place very frequently and often on simple issues such as approving or refusing a budget. Hence, it is not a surprise that a number of cryptographic protocols for boardroom voting have been already proposed. In this work, we introduce a security model adequate for the boardroom voting context. Further, we evaluate the efficiency of four boardroom voting protocols, which to best of our knowledge are the only boardroom voting protocols that satisfy our security model. Finally, we compare the performance of these protocols in different election settings.**

I. Introduction

In many practical applications of cryptographic protocols, efficiency is one of the most crucial aspects to consider. Electronic voting, where cryptographic protocols are deployed to fulfill multiple security requirements such as vote secrecy or integrity, is not an exception. Currently, electronic voting has become of paramount interest for various contexts reaching from political elections at polling stations [1] or over the Internet [2]–[4][1] to board elections in organisations such as universities, companies, or associations [5]. A new and promising application area of electronic voting is boardroom voting, which in practice takes place very frequently and often on simple issues such as approving or refusing a budget. Hence, it is not a surprise that a number of cryptographic protocols for boardroom voting have been already proposed [6]–[13]. Generally, such elections are spontaneously initiated and conducted quickly. Therefore, voters use small mobile devices such as their smartphones or tablet computers. Under these circumstances, the efficiency of cryptographic protocols for boardroom voting becomes even more important. The research goal of this work is to evaluate cryptographic protocols for boardroom voting regarding their efficiency within different election settings.

In order to achieve our goal, we proceed as follows. First, we study the boardroom voting context and propose a security model tailored to that context. We review the literature and identify boardroom voting protocols that satisfy the proposed security model. We evaluate the identified protocols with regard to their efficiency. This evaluation is divided into two steps: First, we decompose boardroom voting protocols in their cryptographic building blocks and determine the required number of modular exponentiations. Second, we provide parametrized efficiency functions for the boardroom voting protocols. Finally, we compare the performance of boardroom voting protocols within different election settings, i.e., by looking at different electorate sizes and ballot types. [2]

II. Boardroom Elections

In this section we provide a general overview of elections in the context of boardroom voting. We mainly focus on the setting and procedure of boardroom elections. Further, we consider boardroom voting elections in the context of companies, where current security mechanisms and policies are deployed.

Electorate: Boardroom elections, which usually take place during boardroom meetings, are small-scale elections and therefore have a limited electorate. We denote the number of voters in the electorate as N. Some of them may follow the boardroom meeting and participate in the voting process from a remote place, for instance via phone or video conference.

Type of Election: Boardroom voting is mostly used for simple issues such as yes/no decisions, i.e., the voting rules are relatively simple and the number of voting options is limited. A voting rule, which we consider in our analysis, is where voters can select up to one of L options.

Equipment and Communication Infrastructure: Boardroom elections are often spontaneously initiated and conducted, and voters usually bring their commonly used device, for example a notebook, smartphone or tablet computer, on which the voting application has already been installed. Additionally, the voters' devices are usually connected to a common network, which allows exchanging messages (encrypted and signed votes) via a reliable broadcast channel.

Public-Key Infrastructure: In order to ensure the authenticity and integrity of the exchanged messages, boardroom

[1] The Norwegian government discarded Internet voting in 2014.

[2] An extended version of this paper is published at http://eprint.iacr.org/2015/558.

This work has been published in Proceedings of the 10th International Conference on Availability, Reliability and Security (ARES), 2015 , p. 224-229, IEEE, August 2015. DOI: 10.1109/ARES.2015.75

elections imply the existence and deployment of a public-key infrastructure (PKI).[3]

III. Research Method

The efficiency of boardroom voting protocols is essentially determined by the computational complexity of the used cryptographic primitives/protocols. In order to avoid that security is sacrificed for the sake of efficiency, the considered protocols should fulfill a security model adequate to the boardroom context. We therefore propose an adequate security model, which consists of security requirements and adversarial capabilities. We identify boardroom voting protocols from the scientific literature and consider only those that satisfy the proposed security model. Further, we determine the efficiency of each protocol by calculating the number of exponentiations of all cryptographic building blocks in use. For the sake of comparison, in our analysis we omit exponentiations with small exponents: If calculations occur in a multiplicative group of order q, and l_q denotes the bit length of q, we consider only exponentiations for exponents with the potential of having the bit length of l_q. Finally, we provide a performance comparison of all considered protocols within different election settings[4]

IV. Adequate Security Model

In this section we propose an adequate security model for context of boardroom voting. The security model consists of two parts, namely the security requirements and the adversarial capabilities. Further, we advocate the adequacy of the security model by justifying the adversarial capabilities.

A. Security Requirements

Many scientific works, e.g. [15]–[18], are dedicated to the definition of security requirements in the context of Internet voting protocols. As the requirements eligibility and anonymity depend on the specific implementation in terms of authentication, within this work we restrict our attention to secrecy, integrity, robustness. We build our work upon the following definitions:

Secrecy: The protocol does not provide more evidence about an eligible voter's intention than the election result [15].

Integrity: The protocol ensures that each vote is correctly included in the election result [15].

Robustness: The protocol returns the election result [19].

B. Adversarial Capabilities

We assume following restrictions on the adversary[5]:

The adversary is computationally restricted and therefore cannot break the underlying cryptographic primitives/protocols, namely the decisional Diffie-Hellman assumption holds. We justify this restriction by the fact that boardroom elections are spontaneously initiated and conducted, i.e. the time frame to manipulate votes is relatively limited. Furthermore, the impact of the election result is timely limited, i.e. the violation of vote secrecy in the long-term is not a significant concern.

The adversary is not able to corrupt more than half of the voters with regard to secrecy violations. We justify this assumption by the fact that if more than half of the voters are corrupt, they can dictate their intention regarding the election outcome. In this case vote secrecy becomes subordinated (relatively irrelevant).

The adversary cannot compromise or coerce voters to violate their vote secrecy[6]. Hence, we assume that voters are honest and do not provide the adversary with any proof of how they voted. Further, voters are free to ignore adversarial instructions, which aim to violate their vote secrecy. Similarly to Benaloh [20] , we justify this restriction by the fact that modern-technology, for instance wearable cameras such as Google glasses, render even protections used in the paper-based polling station elections mostly inefficacious.

The adversary is not able to disrupt messages sent over the communication channel. This restriction is justified by the fact that mechanisms that ensure a reliable communication channel, for instance Byzantine agreement [21], are in place.

The adversary is not able to block at least half of the voters from the communication channel. This restriction is justified by the fact that even if the adversary can block one communication channel, e.g. WLAN, voters can use a different channel, e.g. mobile network.

The adversary is not able to compromise voters' devices in order to violate vote secrecy. We justify this restriction by the fact that in the business context, there are often policies in place, which mitigate the risk of malware infection.

V. Literature Review

A number of boardroom voting protocols have been proposed in the literature. A seminal work to boardroom voting protocols has been presented by DeMillo et al. [6] and extended in [7]. The protocol uses a decryption mix net approach for anonymizing votes. A second branch of works has been initiated by Kiayias et al [8] and improved in [9], [10]. All of these protocols rely on the availability of all voters to compute the election result, thus failing to fulfill robustness according to our proposed security model. Ritter [11] adapted the technique of homomorphic tallying to the boardroom voting context and proposed two derivations of his protocol tailored towards simple and complex ballots. In similar vein, Kulyk et al. [12] adapted the technique of reencryption mix net-based election to the boardroom voting context. Further, Khader et al. [13] suggested a boardroom voting protocol building upon self-dissolving commitments. These protocols, namely [11], [12] and [13] satisfy the proposed security model and are therefore considered in the remainder of this work.

[3] Note that if the PKI does not exist in beforehand, it can be setup using a protocol such as SafeSlinger [14].

[4] We limit our analysis to the theoretical performance only, since the performance in practice depends largely on the implementation details.

[5] In the remainder of this work, we do not distinguish between adversarial actions and benign failures.

[6] We recognize that there might be scenarios where this assumption cannot be made, and consider the forms of coercion that are possible in boardroom voting setting and the extent to which the existing solutions can ensure coercion resistance a direction in future work

TABLE I: A summary of the computational costs of the cryptographic primitives used in this paper in terms of number of exponentiations.

Cryptographic Primitive	*Task*	*Exponentiations*
ElGamal cryptosystem	Encryption	2
	Decryption	1
Digital signature algorithm (DSA)	Generate signature	1
	Verification	2
Advanced encryption standard	Encryption	0
	Decryption	0
Diffie-Hellman key exchange	Calculating own part of key	1
	Combining with the part from communication partner	1
Commitment	Commitment	1
	Check opening	1
Knowledge of discrete logarithm	Proof generation	1
	Verification	2
Equality of discrete logarithms	Proof generation	2
	Verification	4
Well-formedness proof for 2 options [22]	Proof generation	6
	Verification	8
Well-formedness proof for $L > 2$ options [23]	Proof generation	11
	Verification	11
Threshold ElGamal key generation for $t \leq N$	Generate public key	$2N$
	Generate private key shares	$N + t - 2$
Threshold ElGamal decryption for $t \leq N$	Partial decryption	1
	Proof generation	2
	Verification	$4\,t$
	Plaintext reconstruction	t
Re-encryption mix net (N encryptions)	Mixing	$2N$
	Proof generation	$8N + 5$
	Verification	$9N + 11$

VI. Cryptographic Building Blocks of Boardroom Voting Protocols

After the protocols that satisfy our security model have been identified, we decompose the protocols into their cryptographic building blocks. The results are presented in Table I.

ElGamal cryptosystem: Generally, boardroom voting protocols build upon the ElGamal cryptosystem [24], which is of particular interest due to its homomorphic properties.

Digital signatures: The sender authenticity is ensured by the use of digital signatures. For the sake of consistency in our efficiency evaluation, we assume the use of DSA signatures [25] as these signatures build upon the DDH assumption.

Advanced encryption standard: Given the fact that symmetric cryptosystems are generally more efficient than their asymmetric counterparts, private channels between communication partners are established by the use of the AES cryptosystem.

Diffie-Hellman key exchange: Building upon an established PKI, the Diffie-Hellman key exchange protocol [26], [27] allows two communication partners to create an AES key.

Commitment schemes: The herein investigated protocols build upon a simple commitment scheme in which a group generator is raised to the power of the committing value, as used in [22], e.g. to commit on a vote in the voting phase which is only revealed in the tallying phase.

Zero-knowledge proofs: The proofs that are used in boardroom voting protocolls are the *proof of knowledge of a discrete logarithm* (also called preimage proofs) [28], the *proof of equality of two discrete logarithms* [29] and *well-formedness proofs* which are used to prove that a ciphertext/commitment contains a message from a given set of messages [22], [23].

Threshold ElGamal key generation and decryption: The herein investigated protocols distribute the trust by generating a public ElGamal key and a set of private ElGamal key shares with a combination of a distributed secret sharing scheme [30] with the ElGamal cryptosystem, as suggested in [31].

Reencryption mix net: The purpose of a reencryption mix net is to anonymize a set of ciphertexts, such that links between the output and the input are removed. The boardroom voting protocols considered in this work implement a correctness proof of shuffle according to [32].

VII. Boardroom Voting Protocols

This section describes and evaluates the four boardroom voting protocols satisfying the proposed security model. Each protocol is described in terms of required calculations per voter (relying on used cryptographic building blocks). Thereby, we are able to provide the overview of protocols' efficiency in form of parametrized functions.[7]

A. Homomorphic tallying, single encryption

The general idea of this protocol is that voters encrypt their ballots into one ciphertext. The individual ciphertexts of voters are aggregated into one by using the homomorphic property of the ElGamal cryptosystem and then tallied.

Initialization: Parts of the protocol rely on the use of private communication channels. Therefore, in the initialization phase, the Diffie-Hellman key exchange is executed between each pair of voters (N exponentiations). Additional costs of each voter are the signing of one message and verifying the signatures of $N - 1$ messages. The total costs of the initilization phase are thus $3N - 1$ exponentiations.

[7] In addition to computational complexity, also communication complexity has an impact on the overall efficiency. This depends however on practical aspects such as protocol implementation and network load.

Setup: In the setup phase, the distributed key generation for the threshold of $t \leq N$ is executed. This takes a total of $3N + t - 2$ exponentiations ($N + t - 2$ for generating private key shares, and $2N$ for generating the public key). Each voter has to sign $N + 1$ and to verify the signatures of $3(N - 1)$ messages, resulting in total computational cost for the setup phase of $10N + t - 7$ exponentiations.

Vote Casting: In order to cast her vote, first the voter encrypts her vote into a single ElGamal ciphertext (2 exponentiations)[8]. In addition to the encrypted vote, voters provide a well-formedness proof stating that the ciphertext contains a valid vote. The proof takes 11 exponentiations. Including the costs of signing one message and verifying the signatures on $N - 1$ messages, the total computational costs for the vote casting phase are $12 + 2N$ exponentiations. *Tallying:* To compute the final tally, first each voter verifies the correctness of all $N - 1$ well-formedness proofs obtained from the others, requiring $11(N - 1)$ exponentiations. The decryption of the election result requires from each voter the computation of a partial decryption share (1 exponentiation) and the generation of a validity proof for the decryption (2 exponentiations), the verification of other voters' proofs (4 exponentiations for each proof), and the reconstruction of the plaintext from the threshold amount of partial decryption shares (t exponentiations). Note, that since a total of t valid partial decryption shares is needed for the plaintext reconstruction, the exact number of proofs one has to verify depends on whether there are corrupted voters that send invalid proofs that still need to be processed. However, the presence of invalid proofs seems not to be realistic because the effort needed for such a corruption is too high compared to the pay-off for the adversary - the election can only be slowed down, but not hindered completely. Therefore in our analysis we consider the case, where no voters send faulty messages, and a total of $t - 1$ proofs need to be verified.

Given the use of the exponential ElGamal cryptosystem, the determination of the final election result requires the computation of a discrete logarithm. The computation of this logarithm is implemented by an exhaustive search over all $\binom{N+L}{L}$ possibilities, resulting in an average of $\binom{N+L}{L}/2$ exponentiations. Including the costs of signing one and verifying the signatures on $t - 1$ messages leads to the total computational costs for this phase of $11N + 7t - 13 + \binom{N+L}{L}/2$ exponentiations.

TABLE II: Homomorphic tallying, single encryption: The amount of exponentiations needed within a protocol runs.

Initialization	$3N - 1$
Setup	$10N + t - 7$
Vote casting	$12 + 2N$
Tallying	$11N + 7t - 13 + \binom{N+L}{L}/2$
Total	$-9 + 26N + 8t + \binom{N+L}{L}/2$

B. Homomorphic tallying, multiple encryptions

Similarly to the homomorphic tallying (single encryption), voters' ciphertexts are aggregated in the tallying phase. As opposed to the single encryption case, in the multiple encryption case, voters encrypt their ballots individually for each possible voting option. Hence, in the tallying phase, the election result is tallied by computing the number of votes each voting option has obtained, individually.

The initialization and setup phase are identical to the single encryption case.

Vote Casting: In this approach, the cast vote is multiple ciphertexts each representing one option. The selected option is represented as an encryption of 1, and all the other options as encryptions of 0, with all the encryptions requiring a total of $2L$ exponentiations. One then has to compute the well-formedness proofs for each voting option as well as for their sum. These proofs require $6(L+1)$ exponentiations. Including the costs for signing one message and verifying the signatures of $N - 1$ messages, the resulting computational costs for the vote casting phase are $8L + 2N + 5$.

Tallying: Before computing the final tally, all $(L+1)(N-1)$ well-formedness proofs are checked by every voter. This requires $8(LN + N - L - 1)$ exponentiations. The threshold decryption is then executed L times, once for each voting option. As in Section VII-A, we assume that all the validity proofs for the decryption sent during this phase are valid, and therefore the threshold decryption requires a total of $5Lt - L$ exponentiations. Calculating the final result from the decrypted values requires the computation of the result for each voting option individually, which, however, consists of exponentiations with small exponents only (with values up to N), and therefore is not included into the analysis. Additional costs consist of signing one message and verifying the signatures of $t - 1$ messages. The final computational costs of the tallying phase are therefore $8LN + 8N - 9L - 9 + 5Lt + 2t$ exponentiations.

TABLE III: Homomorphic tallying, multiple encryptions: The amount of exponentiations needed within a protocol run.

Initialization	$3N - 1$
Setup	$10N + t - 7$
Vote casting	$8L + 2N + 5$
Tallying	$8LN + 8N - 9L - 9 + 5Lt + 2t$
Total	$-12 + 23N + 8LN - L + 3t + 5Lt$

C. Mix Net-Based Tallying

The third protocol satisfying the proposed security model builds upon a different approach to anonymize cast votes. Rather than aggregating votes, in the mix net-based approach votes are anonymized through shuffling and tallied individually. Initialization and setup are identical to the protocol in Section VII-A, thus we omit their description here.

Vote Casting: To cast a vote, the voter first computes an ElGamal encryption of her ballot (2 exponentiations). In addition, the voter proves knowledge of the plaintext to enforce ballot-independence [34], which is done by proving the knowledge of discrete logarithm (1 exponentiation). Additional computational costs occur for signing one message and verifying the signatures on $N - 1$ messages, with the total costs being $2N + 2$ exponentiations.

Tallying: At the beginning of the tallying phase, each voter verifies the plaintext knowledge proofs of all other voters,

[8] We assume that the device used by the voter encrypts the voting option that she intended to cast. This can be ensured with either relying on the trustworthiness of the device, or by applying verification mechanism like the Benaloh challenge [33].

resulting in $2(N-1)$ exponentiations. The remainder of the tallying is executed in two sub-phases: the mixing phase and the decryption phase.

In the mixing sub-phase, a set of voters is selected for acting as a reencryption mix nodes—one after each other according to the order agreed on in the setup phase. At least 2 honest voters must participate in the mixing process; thus, the total amount of mix nodes is $N-t+2$[9]. In each mixing round, the computational costs are $10N+5$ exponentiations for the mix node ($2N$ for reencryption, $8N+5$ for the proof of shuffle), followed by $9N+11$ exponentiations for the rest of the voters verifying the shuffle. Additional costs for the round consist of signing (by the mixing node) and verifying the signature (by the rest of the voters) of one message, consisting of 3 exponentiations. For $N-t+2$ mixing rounds, the total costs for the mixing sub-phase thus are $19N^2+59N-19Nt-19t+36$ exponentiations.

In the decryption phase, each of the N anonymized ciphertexts is decrypted using the threshold decryption, resulting in $5Nt-N$ exponentiations[10]. Adding the costs of signing one message and verifying $t-1$ messages, the total costs of the decryption sub-phase consist of $5Nt-N+2t-1$ exponentiations.

In order to obtain the result, the votes for each option are being added up over all decrypted votes. The total computational costs of the tallying phase are therefore $19N^2+58N-14Nt-17t+35$ exponentiations.

TABLE IV: Mix net-based tallying: the amount of exponentiations needed within a protocol run.

Initialization	$3N-1$
Setup	$10N+t-7$
Vote casting	$2N+2$
Tallying	$19N^2+58N-14Nt-17t+35$
Total	$19N^2+73N-14Nt-16t+29$

D. Protocol Based on Self-Dissolving Commitments

Similar to the protocol in Section VII-A, the fourth protocol is based on the concept of homomorphic tallying, with the vote encoded as a single value.

Setup: Every voter picks a random value and communicates its commitment together with the preimage proof to all other voters, which requires 2 exponentiations for commitment and proof, and 1 exponentiation for signing the message. When all $N-1$ such pairs of commitments and proofs have been received from the other voters and successfully verified ($2N-2$ exponentiations for the verification of the proof, and $2N-2$ for the verification of signatures on $N-1$ messages), each voter uses them to calculate the value needed for casting the vote (note, that this calculation requires only multiplications and calculation of inverse, therefore not included in our efficiency analysis). The total costs of setup phase are therefore $4N-1$ exponentiations.

Vote Casting: To cast a vote for an option, each voter computes one commitment to their vote (1 exponentiation). Before communicating the commitment to everyone, the voter computes the corresponding well-formedness proof (11 exponentiations)[11]. Casting the vote to the others is done in two rounds, first by communicating the proof (which serves the purpose of a commitment) and second by communicating the vote (2 exponentiations for signing 2 messages, $4N-4$ for verifying the signatures on $2(N-1)$ messages). Sending the proof and the vote separately is necessary for the fairness property of the protocol. Overall, the computational costs for the vote casting phase are $4N+10$ exponentiations.

Recovery: If in any previous step a voter stops following the protocol, a special recovery phase exists for the set of remaining voters to guarantee the robustness of the protocol. Given the fact that the adversary might block the access to the communication channel for several voters IV, we consider the conduct of a recovery phase. The phase requires 1 exponentiation in addition to one proof of discrete logarithm equality (2 exponentiation) and verification of $N-1$ such proofs ($4N-4$ exponentiations). Furthermore, additional costs are for signing of one message and verifying the signatures of $N-1$ messages, resulting in total of $6N-2$ exponentiations.

Tallying: After positive verification of the well-formedness proofs exchanged during vote casting ($11(N-1)$ exponentiations), the encoded final result is obtained, which requires an an exhaustive search over $\binom{N+L}{L}$ possibilities, and correspondingly performing an average of $\binom{N+L}{L}/2$ exponentiations.The total costs for tallying phase are therefore $11N-11+\binom{N+L}{L}/2$.

TABLE V: Protocol based on self-dissolving commitments: the amount of exponentiations needed within a protocol run.

Setup	$4N-1$
Vote casting	$4N+10$
Recovery	$6N-2$
Tallying	$11N-11+\binom{N+L}{L}/2$
Total (with recovery round)	$25N-4+\binom{N+L}{L}/2$
Total (without recovery round)	$19N-2+\binom{N+L}{L}/2$

VIII. Efficiency Comparison in Different Election Settings

In this section we illustrate and compare the performance of the protocols in different election settings. For our evaluation, we consider the established PKI based on DSA keys. In addition, according to the security model proposed in Section IV, we set the threshold value $t=\lfloor N/2\rfloor+1$.

The evaluation results show that the protocols based on self-dissolving commitments and single encryption homomorphic tallying perform particularly well for simple ballot types, for instance with 2 voting options. The multiple encryption homomorphic tallying protocol outperforms the self-dissolving commitments and the single encryption homomorphic protocol when the election setting gets more complex. More precisely, having 7 or more voters with 5 voting options; having 4 or more voters with 10 voting options; and having 2 or more voters with 30 voting options. In case of 30 voting options,

[9] Note, that the verification of shuffle correctness is still being performed by all the voters, so that each voter can verify the integrity of the election without trusting anyone else.

[10] Similar to Sections VII-A and VII-B, we assume that no voters send faulty validity proofs for the decryption.

[11] This is possible considering the commitment in setup round and the commitment in this round as an ElGamal-like ciphertext.

for less than 25 voters, the mix net-based protocol outperforms all other protocols.

IX. Conclusion

In this work, we conducted an efficiency evaluation of cryptographic protocols proposed in the literature for boardroom voting setting. For this, we first derived a security model adequate to the boardroom voting setting. Then, we conducted a literature review on boardroom voting protocols, and extracted those that satisfy our security model. Further, we analyzed the efficiency of each protocol by calculating the number of exponentiations of the used cryptographic building blocks and provide an overview of the protocols' efficiency in form of parametrized functions. Finally, we provided a performance comparison of the considered protocols within different election settings. The results of the performance comparison indicate that there is no protocol that is the most efficient in all election settings. Rather, choosing the most adequate protocol depends on the concrete election setting.

Acknowledgment

This project (HA project no. 435/14-25) is funded in the framework of Hessen ModellProjekte, financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

Hom. tallying, single encryption	$-9+26N+8t+\binom{N+L}{L}/2$
Hom. tallying, self-dissolving comm.	$25N-4+\binom{N+L}{L}/2$
Hom. tallying, multiple encryptions	$-12+23N+8LN-L+3t+5Lt$
Mix Net	$19N^2+73N-14Nt-16t+29$

References

[1] J. Pomares, I. Levin, R. M. Alvarez, G. L. Mirau, and T. Ovejero, "From piloting to roll-out: voting experience and trust in the first full e-election in argentina," in *6th International Conference on Electronic Voting: Verifying the Vote, EVOTE 2014*. IEEE, 2014, pp. 1–10.

[2] Estonian National Electoral Committee, "E-Voting System General Overview," 2010. [Online]. Available: http://www.vvk.ee/public/dok/General_Description_E-Voting_2010.pdf

[3] "Switzerland: Online voting," https://www.ch.ch/en/online-voting/, [Online; accessed 13-March-2015].

[4] O. Spycher, M. Volkamer, and R. Koenig, "Transparency and technical measures to establish trust in norwegian internet voting," in *E-Voting and Identity*. Springer, 2012, pp. 19–35.

[5] IACR, "IACR Election 2013," http://www.iacr.org/elections/2013/, 2013, [Online; accessed 2-March-2015].

[6] R. A. DeMillo, N. A. Lynch, and M. J. Merritt, "Cryptographic protocols," in *Proceedings of the fourteenth annual ACM symposium on Theory of computing*. ACM, 1982, pp. 383–400.

[7] A. Alkassar, R. Krimmer, and M. Volkamer, "Online-wahlen für gremien," *DuD Datenschutz und Datensicherheit*, vol. 8, no. 29, 2005.

[8] A. Kiayias and M. Yung, "Self-tallying elections and perfect ballot secrecy," in *Public Key Cryptography*. Springer, 2002, pp. 141–158.

[9] J. Groth, "Efficient maximal privacy in boardroom voting and anonymous broadcast," in *Financial Cryptography*. Springer, 2004, pp. 90–104.

[10] F. Hao, P. Y. Ryan, and P. Zieliński, "Anonymous voting by two-round public discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–67, 2010.

[11] J. Ritter, "Decentralized e-voting on android devices using homomorphic tallying," Master Thesis, Bern University of Applied Sciences, Biel, Switzerland, 2014.

[12] O. Kulyk, S. Neumann, C. Feier, M. Volkamer, and T. Köster, "Electronic voting with fully distributed trust and maximized flexibility regarding ballot design," in *6th International Conference on Electronic Voting (EVOTE)*. IEEE, Oct. 2014, pp. 1–10.

[13] D. Khader, B. Smyth, P. Y. Ryan, and F. Hao, "A fair and robust voting system by broadcast," in *EVOTE'12: 5th International Conference on Electronic Voting*, 2012.

[14] M. Farb, Y.-H. Lin, T. H.-J. Kim, J. McCune, and A. Perrig, "Safeslinger: Easy-to-use and secure public-key exchange," in *MobiCom'13, 19th Annual International Conference on Mobile Computing & Networking*, Miami, USA, 2013, pp. 417–428.

[15] S. Neumann and M. Volkamer, *A Holistic Framework for the Evaluation of Internet Voting Systems*, ser. Design, Development, and Use of Secure Electronic Voting Systems. IGI Global, 2014, ch. 4, pp. 76–91.

[16] M. Volkamer and R. Vogt, "Basic set of security requirements for Online Voting Products," Tech. Rep. BSI-PP-0037, 2008, common Criteria Protection Profile.

[17] Council of Europe, "Legal, Operational and Technical Standards for E-Voting. Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe and explanatory memorandum," Council of Europe Publishing, 2004.

[18] M. Volkamer and R. Grimm, "Determine the Resilience of Evaluated Internet Voting Systems," in *First International Workshop on Requirements Engineering for e-Voting Systems*, ser. RE-VOTE '09. IEEE Computer Society, 2009, pp. 47–54.

[19] F. Shirazi, S. Neumann, I. Ciolacu, and M. Volkamer, "Robust electronic voting: Introducing robustness in civitas," in *Requirements Engineering for Electronic Voting Systems (REVOTE), 2011 International Workshop on*. IEEE, 2011, pp. 47–55.

[20] J. Benaloh, "Rethinking voter coercion: The realities imposed by technology," *The USENIX Journal of Election Technology and Systems*, vol. 82, 2013.

[21] M. Correia, L. C. Lung, N. F. Neves, and P. Veríssimo, "Efficient byzantine-resilient reliable multicast on a hybrid failure model," in *Reliable Distributed Systems, 2002. Proceedings. 21st IEEE Symposium on*. IEEE, 2002, pp. 2–11.

[22] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *European transactions on Telecommunications*, vol. 8, no. 5, pp. 481–490, 1997.

[23] J. Groth, "Non-interactive zero-knowledge arguments for voting," in *Applied Cryptography and Network Security*. Springer, 2005, pp. 467–482.

[24] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*. Springer, 1985, pp. 10–18.

[25] P. FIPS, "186-4. digital signature standard (dss)," *National Institute of Standards and Technology (NIST)*, 2013.

[26] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[27] A. Menezes and B. Ustaoglu, "On reusing ephemeral keys in diffie-hellman key agreement protocols," *International Journal of Applied Cryptography*, vol. 2, no. 2, pp. 154–158, 2010.

[28] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in cryptology—CRYPTO'89 proceedings*. Springer, 1990, pp. 239–252.

[29] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Advances in Cryptology—CRYPTO'92*. Springer, 1993, pp. 89–105.

[30] T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology—EUROCRYPT'91*. Springer, 1991, pp. 522–526.

[31] V. Cortier, D. Galindo, S. Glondu, M. Izabachene *et al.*, "A generic construction for voting correctness at minimum cost-application to helios." *IACR Cryptology ePrint Archive*, vol. 2013, p. 177, 2013.

[32] B. Terelius and D. Wikström, "Proofs of restricted shuffles," in *Progress in Cryptology–AFRICACRYPT 2010*. Springer, 2010, pp. 100–113.

[33] J. Benaloh, “Simple verifiable elections,” in *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop*. USENIX Association, 2006, pp. 5–5.

[34] B. Smyth and D. Bernhard, “Ballot secrecy and ballot independence coincide,” in *ESORICS*, 2013, pp. 463–480.