athematical Problems of Computer Science 36, 75–82, 2012.

A New Approach to Receipt-Free E-Voting

Aram Jivanyan

Russian-Armenian (Slavonic) University
e-mail: jivanyan@gmail.com

Abstract

In this paper we introduce a new electronic voting protocol with provable security properties. The generic method of voting presented here allows choosing different approaches for ballot casting to provide the most user-friendly interface.

Introduction

ectronic voting schemes found in the literature can be classified by their approaches into the llowing three categories:

- Schemes using blind signature;
- Schemes using homomorphic encryption;
- Schemes using mix networks;

oting schemes based on blind signature are simple, efficient, and flexible, but they require an onymous channel. Frequently, the anonymous channel is implemented using a mixnet, but if a cure mixnet is available, a blind signature is no longer required. Voting schemes based on momorphic encryption are efficient at the initial stage, but intensive zero-knowledge proofs e used to prove the validity of each ballot in the voting stage, which are costly for the voters. oting schemes based on mixnet are generally not efficient because they require a huge amount computing for multiple mixers to prove the correctness of their mixing. However, the recent sults of [8], [9], [10], and [11] have greatly improved the efficiency of mixnet. In electronic ting schemes based on mix networks the voter typically interacts with the voting machine to nerate an encryption of his plaintext ballot. But the voter should not trust the voting machine correctly capture his vote. The voting machine has to prove a voter at the ballot casting phase it the encrypted ballot accurately reflects the voter's intent. At the same time the protocol ould be receipt-free. This means that nothing should be given to voters while providing such a of that could reveal the voter's choice. Earlier studies in cryptographic voting schemes sumed that each voter would have a trusted device to perform cryptographic operations at the ting place. This assumption is not realistic and therefore an active research is going to develop r-friendly voting schemes requiring minimal efforts from voters at polling stations to vote. e focus on cryptographic electronic voting schemes which do not require paper-based sub-dules. Recently an electronic receipt-free voting scheme based on mix networks was proposed Adida and Neff. The scheme provides high soundness level but it is not efficient for tallying. e ballots in that scheme consist of M ciphertexts, where M is the number of candidates in ction . Thus the mix process cost grows linearly depending on the number of candidates. The h cost of the mix process remains the weak point of mixnet-based voting schemes. Our

motivation comes from the point of constructing another electronic voting scheme which will own all the desirable security properties and be efficient at the same time.

We propose a novel electronic voting scheme based on mix networks. While providing a lower level compared with Adida's scheme, our approach yields to very effective mix process and at the same time provide receipt-freeness. The idea is as follows: each voter before choosing his candidate is required to input l secret keys, where l is a system parameter of the elections. After entering the secrets, the voting machine encrypts the inputted secrets by an election public key. Being convinced that all the secrets are already encrypted, the voter randomly chooses one of the secrets and challenges the voting machine to provide special proofs, that the remaining secrets are encrypted correctly. The voting machine uses Fiat-Shamir hash trick [2] to provide non-interactive zero-knowledge proofs of correct encryption. After that the voter selects his candidate and uses the single unopened secret key to encrypt his vote. We will specify later what encryption method the voter will use for this step. Thus if the voting machine did not capture all the secrets properly, it would be detected with $1-1/l$ probability, because all those proofs could be verified outside the voting place.

2. Modeling

We assume that voters vote at certified polling stations and there is one election race with M candidates. As in most election schemes, we also include the notion of "bulletin board" ***BB*** as a shared memory where all the authorized parties have a sequential access to writing after authentication and any observer has a read access. At any moment it must be accessible both for writing and reading via election website. We identify each of the following players with a fixed role he must play during the election process:

- **Polling Stations**: For simplicity we assume the existence of only one Polling Station (PS) with one voting machine which can be malicious. We also assume that PS coordinates all the electoral process not to burden our scheme with additional players. When the election begins, PS collects all the votes and posts them to ***BB***. It is assumed that only legitimate voters will be allowed by PS to vote and they will be able to vote only once. PS owns a certified public/private key pair for digital signing of each voter's receipt given by the voting machine.
- **Decryption parties**: There are K decryption parties denoted by set $\{D_1, D_2, ..., D_k\}$ who own the shares of decryption key needed for ballots decryption. Each decryption party can post to ***BB***.
- **Mix parties**: Mix party performs the mixing of encrypted votes. Any certified participan can act as a mix party and there can be as many mix parties as needed. We will assume that at least one of these parties is honest.
- **Voters**: There are N voters $\{V_1, V_2, ..., V_N\}$ who cast a vote.
- **Verifiers**: A verifier can be anyone who is able to perform more complex mathematica operations. Usually a verifier owns some trusted computer device and can perform cryptographic operations outside PS. A verifier can read the ***BB*** and any voter's receipt t check the validity of signature on it, ensure in correctness of vote encryption, as well a verify any proof provided by the voting machine or decryption parties. A verifier ca complain in case he detects an error.

The election process consists of the following phases:

- **Set up**: At this phase the PS initializes the ***BB***, publishes the characteristics of a grou where all public keys used during the election are defined, and the description of a secur

hash function H which is used at receipt preparation. All decryption servers and the Polling Station publish their certified public keys. The candidate name encoding is also defined and announced to all parties.

- **Voting**: At this phase PS authenticates each voter and checks if he/she has already voted. Then each voter votes using a voting machine according to the fixed protocol which will be described later and gets a special receipt signed by PS, after which his/her ballot will be posted to ***BB***.
- **Vote mixing**: Each mix party takes the encrypted ballots from the ***BB***, re-randomizes them and posts the resulted ballots in shuffled order again to ***BB*** for the next mix or the decryption party to perform respective actions. A mix party can also be required to provide additional data for verification purposes.
- **Tallying**: At this phase the encrypted ballots are opened by decryption parties. Each decryption server except performing his suited actions also provides some additional data required for a verification of its actions. The output of this final phase will be an anonymous list of casted ballots to be already counted.

Next we present all the phases in more details.

1 Generation of Voting Parameters

At this phase the Polling Station initializes the BB and publishes the parameters of a group, where all public keys used for elections are defined. These are two large primes P and Q where P = 2Q+1. All public keys are defined over the Q order subgroup G_Q of Z_P. PS publishes public key. The key pair of the Polling Station is (PK_{PS}, SK_{PS}). The decryption servers generate the election public key Y using the threshold El-Gamal [1], such that each decryption party has a secret share SK_l. A cryptographically secure hash function H definition used in election is specified.

Assume that there are M candidates. Let's take an M-order abelian group $(M, \circ)$. The group specification will be given later: on the whole it does not matter in the building of our e-voting protocol. Then each candidate is represented via a unique group element from M. On the other hand, as El-Gamal cryptosystem is used to encrypt all the secret information concerning voter choice, then each element of M should also be represented via some element from G_Q before being encrypted. This can be done in many different ways and again the specific correspondence will be given later.

2 Voting

We do not discuss here the authentication phase of voters and vote unicity checks done by PS. Instead we detail how an authenticated and first time voting voter casts his/her vote. Each voter is given an electronic ballot with a unique ID. Then he uses the voting machine to cast his vote. While voting the voter should follow the specific steps of a certain voting protocol used in election. There are a number of electronic voting protocols proposed in the literature [1], [2], [3].

The intuition for our voting protocol looks as follows: the voter first enters l secret keys voted by $\{k_1, k_2, \ldots, k_l\}$, which are elements of our specified group M. The keys are subjected to encryption by the election public key. After getting a proof that all the keys are encrypted, the voter chooses one of the entered keys to keep it secret and opens the remained ones. The opened keys will allow the voter to check later that the encryption of keys is properly entrusted to the

voting machine. This reminds of the simple cut-and-choose approach. But note that all the encryption operations required to be performed for voting are fulfilled by the voting machine until the voter makes his choice. Then the voter uses the unopened key to encrypt his initial choice. Without loss of generality we can assume that k_1 is the key kept secret. If the candidate chosen by the voter is $C \in M$, then the voter simply computes $R = k_1 \circ C$, where $\circ$ is the group operation. Note that no information can be extracted from R about either the voter's choice C or his secretk_1 due to the main properties of algebraic groups. The protocol details will be given in the next section.

2.3 Vote Mixing

Vote mixing is a core step in e-voting process based on mix-networks. Mixing guarantees the anonymity of votes. The mix process is run by N mix servers, where the i-th mix server takes the list of N input ciphertexts $(C_1^i, C_2^i, \dots, C_N^i)$, re-encrypts by using secret randomization factors $(r_1, r_2, \dots, r_N)$ and outputs the list of re-encrypted ciphertexts in randomly permuted order - $(C_{\pi(1)}^i, C_{\pi(2)}^i, \dots, C_{\pi(N)}^i)$. The output list can serve as an input for the next mix server or can be passed to the decryption servers in case when all the mix servers have been proceeded. The mix server can be also required to provide an addition information to prove that

$$\forall i, l \exists j : C_i^l = C_j^{l+1} \&\& \; j = \pi(i).$$

2.4 Tallying

After the mix process when all the ballots are already anonymized, the quorum of decryption parties jointly recover the election private key Y. Then they jointly decrypt all the encrypted ballots by providing special proofs of correct decryption. After this the votes can be counted.

3. Voting Protocol

Before exposing our voting scheme some basic cryptographic primitives should be described in more details. First the encryption algorithm used for encrypting votes and/or voter's secrets and also the augmented primitive are applied for proving the correctness of encryption. We use El-Gamal cryptosystem which is the most usable cryptosystem for e-voting systems due to its properties. Next we introduce the Chaum-Pedersen zero-knowledge proof technique used to prove that the prover knows the secret used for encryption.

El-Gamal cryptosystem:

We start with a description of the El-Gamal public-key cryptosystem [1], and discuss some of its properties, which make this cryptosystem very useful for voting protocols. Let P and Q be two large primes such that $P = 2Q + 1$. We denote by $G_Q$the subgroup of $Z_P$of order Q. All the subsequent arithmetic operations are performed in modulo P unless otherwise stated. Let g be a generator of G_Q. The private key is an element $x \in Z_P$, and the corresponding public key is $y = g^x$. To encrypt a plaintextm $\in G_Q$, a random element $r \in G_P$is chosen and the cipher text is computed as $E_Y(m; r) = (g^r;) = (G;M)$, so the El-Gamal ciphertext is a pair of elements of G_Q.

e decryption $D_X(G;M)$ of an El-Gamal ciphertext (G;M) can be computed by the owner of ret x in the following way: $D_X(G;M) = \frac{M}{g^x} = m$.

The next algorithm necessary to build our system is Chaum-Pedersen zero-knowledge proof equality of discrete logarithms. The players are the prover and the verifier. Public inputs are v, v, w where the Prover must prove the Verifier that he knows the discrete logarithm x without evealing it.

Prover: Computes:

- $x \leftarrow Z_Q$
- $a = f^x \bmod p$
- $b = h^x \bmod p$
- $c = hash(v, w, a, b) \bmod q$
- $y = (x + cr) \bmod q$

Prover → Verifier: a, b, c, y

Verifier: Verifies that $f^y \equiv av^c (\bmod p)$ and $h^y \equiv bw^c (\bmod p)$

e Chaum-Pedersen proof thus represents itself as a tuple (a, b, y).

llot-Casting Scheme.

w we present our ballot-casting protocol in more details.

- The voting machine generates random values $(r_1, r_1^{CP}, r_2, r_2^{CP}, r_l, r_l^{CP}) \in Z_Q^{2l}$. It is assumed that the voting machine runs a trusted pseudo-random number generator. Here r_l will be used to encrypt the i-th secret entered by the voter and r_l^{CP} will be used for generating a Chaum-Pedersen proof of correct encryption.
- The voting machine computes and prints on the voter receipt the value
$$H1 = hash\left(g^{r_1} || g^{r_2} || \dots . g^{r_l} || g^{r_1^{CP}} \dots g^{r_l^{CP}}\right)$$
This step is important to covert channels in the receipt. While selecting the random values and committing to them in advance the voter enters any secret information, the voting machine will not be able to reveal any information about the voter's choice by using secret revealing random values.
- The voter inputs l secrets denoted by the set $\{k_1, k_2, \dots, k_l\}$. Each k_l is a randomly chosen element of the group M.
- The voting machine encrypts all the secrets by using the chosen random values. So for each key k_l the ciphertext $E_l = (g^{r_l}, k_l y^{r_l})$ is computed and the Chaum-Pedersen proof $CP_l(g^{r_l^{CP}})$ is generated.
- The voting machine computes and prints on the voter receipt $H2 = hash(E_1 || E_2 || \dots . | E_l || CP_1 || \dots || CP_l)$.
- After seeing the hash value printed on the receipt the voter chooses randomly one of the secrets to keep it secret. The chosen secret will be used to encode the voter choice.
- The voting machine prints all the remained $l - 1$ secrets on the receipt and the voter verifies that all his entered secrets except the last chosen one are properly printed on the receipt.
- The voter makes his choice. Assume that the chosen candidate code is C.
- The voting machine computes and prints on the receipt the value $R = S * C$. The value R is also added to the ballot.

4. Security Properties of the Proposed Protocol

Next we present proof sketches for the main security properties that each voting protocol mus ensure.

4.1 Resistance to Covert Channel

The resistance to the covert channel is a general problem specific to all the voting protocol running on the entrusted voting machine. The problem is that the voting machine can easily select ciphertexts so as to reveal information about the voter secrets, be the voter choice o another kind of must-to-be-kept secret information. Our solution is simple, although not applie yet. As the voting machine encrypts all the voter secret information by using El Gama encryption, we want the voting machine to commit all the randomness he must use before th voter enters any secret information. Our electronic ballot will consist of l El-Gamal ciphertext and l Chaum-Pedersen proofs. The first element of each El-Gamal ciphertexts and of Chaum Pedersen triplets can be viewed as a commitment of random values $H1 = hash(g^{r_1} \,||\, g^{r_2} \,||\, \ldots . g^{r_l} \,||\, g^{r_1{}^{CP}} \ldots g^{r_l{}^{CP}})$The value printed on the receipt at the firs step gives a possibility to verify later that all preselected random values are used so all th ciphertexts are predictable in some sense. Having the g^r and Chaum Pedersen proofs, one ca simply check that all valid ciphertexts are generated with respect to g^r. This means that th corrupted voting machine can no longer use the randomness to covertly reveal any secre information. This proves the covert-channel resistance of our proposed voting protocol.

4.2 Soundness

For soundness we assume that every trustee, every voter except those voting at the certai moment as well as the voting machines are adversarial. Still, if the voter's ballot is encrypte incorrectly, we want either the voter or the helper to catch the problem with reasonably hig probability. The voting machine can cheat by encrypting the secrets incorrectly and trying to foo either the voter or the helper. The voting machine can fool the voter by incorrectly computing th addition of the voter's chosen candidate code and the voter's still-kept-secret key. We assum that an interface-based tool allows the voter to check at Polling Station the correctness o computed expression$R = S \circ C$, where S is the voter's single unopened key and C is the voter' chosen candidate code. While trying to fool the helper, the voting machine can cheat only if i guesses the voter's choice secret key ahead of time. As all the other ciphertexts except the on chosen by the voter are subjected to disclosure, an incorrect encryption of the remained keys wil be detected with an overwhelming probability. So the voting machine has a $1/l$ chance o correctly guessing the key which will remain secret, which means our protocol is at least 1-1/ sound.

Note that the final vote is composed of the voter's secret key and the candidate's code chosen by him. While guessing the correct secret key to be kept secret and encrypting it incorrectly, th voting machine can change the vote only in an unknown way, as it does not know the voter' preferred candidate's code. This means, that our protocol allows the voting machine to chea only "blindly" and not be discovered with probability $1/l$.

3.3 Receipt-freeness

Receipt-freeness ensures that the voter cannot provide any receipt proving the content of his vote, even if the voter wants it. The receipt freeness of our protocol is based on the fact of covert-channel resistance and also the algebraic properties of the used mathematical primitive. As long as the discrete logarithm problem is unsolved and El-Gamal ciphertexts cannot reveal any information about their enclosed messages, the only information from the receipt can be extracted from the value $= S \circ C$. Due to the properties of an algebraic group, we can say that $\forall S' \in M \ \exists \ C' \in M \, s.t. \ S' \circ C' = S \circ C$. So the value R provides unconditional privacy of the voter's choice.

4. Conclusion

We proposed a new ballot casting scheme where each voter encrypts his vote with his secret key with minimal computations. The proposed general method allows the e-voting software vendors to choose the most user-friendly interface. For example, all the secrets and the candidate encodings can be Boolean vectors and the group operation can be simple XOR-ing. Note that XOR can be checked very easily by using visual cryptography methods [4, 20]; or the group Z_m can be chosen for encoding candidates, where m is the number of candidates. Then the group operation which will be also used for vote encryption by users, will be the (+ mod m) operation. The proposed scheme is very effective for ballots mixing and vote counting compared with the other e-voting schemes [18],[19] proposed recently.

References

[1] T. El-Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *Advances in Cryptology - CRYPTO '84, LNCS 196, pp 10-18, 1984*

[2] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems", *Advances in Cryptology-CRYPTO '86, vol. 263 of LNCS*

[3] D. Chaum and T. Pedersen, "Wallet databases with observers". *In Proc. of Crypto'92, Springer-Verlag*, pp. 89-105, 1993. LNCS 740.

[4] D. Chaum, "Secret Ballot Receipts and Transparent Integrity Better and less-costly electronic voting at polling places", *http://vreceipt.com/article.pdf, 2004.*

[5] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, 24(2), pp. 84-88, 1981.

[6] A. Shamir,"How to share a secret", *Communications of the ACM*, 22(11), pp. 612-613, 1979.

[7] D. Boneh, P. Golle, "Almost Entirely Correct Mixing With Application to Voting."

[8] A. Neff. "A verifiable secret shuffle and its application to E-Voting", *In Proc. of ACM CCS'01, ACM Press*, pp. 116-125. 2001.

[9] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle", *Proc. of Crypto '01,Springer-Verlag, LNCS 2139*, 2001.

[10] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, A. Juels, "Optimsitic mixing for exit-polls", *Advances in Cryptology (Asiacrypt 2002)', Vol. 2501 of LNCS, Springer-Verlag, 2002.*

[11] M. Jakobsson, A. Juels and R. Rivest. "Making mix nets robust for electronic voting by randomized partial checking", *In Proc. Of USENIX'02.*

[12] D. Wikstrom. "Four practical attacks for "optimistic mixing for exit-polls"", *Technical Report T2003-04, Swedish Institute of Computer Science, 2003.*
[13] C. Park, K. Itoh and K. Kurosawa. "Efficient anonymous channel and all/nothing election Scheme."*In Proc. of Eurocrypt 93, Springer-Verlag, LNCS765,* pp. 248-259, 1993.
[14] W. Ogata, K. Kurosawa, K. Sako and K. Takatani. "Fault tolerant anonymous Channel". *In Proc. of ICICS 97,* LNCS 1334, pp. 440-444, 1997.
[15] K. Sako and J. Kilian. "Receipt-free mix-type voting scheme". *In Proc. of Eurocrypt 95, Springer-Verlag, LNCS 921, 1995.*
[16] M. Abe. "Universally verifiable mix-net with verification work independent of the number of mix-servers", *In Proc. of Eurocrypt 98, Springer-Verlag,* pp. 437-447, 1998.
[17] T. Moran, M. Naor. "Receipt-free universally-verifiable voting with everlasting privacy."*In Advances in Cryptology CRYPTO 2006.*
[18] B. Adida and C. A. Neff. "Efficient receipt-free ballot casting resistant to covert channels."*In EVT/WOTE, 2009.*
[19] M. Stadler "Publicly verifiable secret sharing". *Proc. Eurocrypt '96,* pp. 190-199. 1996.
[20] M. Naor and A. Shamir. "Visual cryptography". *In EUROCRYPT,* pp 112,1994.

Գաղտնիություն ապահովող էլեկտրոնային քվեարկության նոր մոտեցում

Ա. Ջիվանյան

Ամփոփում

Մենք ներկայացնում ենք էլեկտրոնային քվեարկության նոր համակարգ անվտանգության ձևական ապացուցվող հատկություններով: Առաջարկված է քվեարկության ընդհանուր մոտեցում, որը թույլ է տալիս ընտրել վերջնական քվեարկության տարբեր մեթոդներ, ինչը հնարավորություն է տալիս ընտրե օգտագործողի տեսակետից ամենահարմար ինտերֆեյսը:

Новый подход к обеспечению секретного электронного голосования

А. Дживанян

Аннотация

Мы представляем новую систему электронного голосования с формально доказуемыми свойствами безопасности. Предложен общий подход, позволяющий использовать различные конечные методы, что дает пользователям свободу выбора наиболее подходящего для них интерфейса.