Providing Receipt-freeness in Mixnet-based Voting Protocols

Byoungcheon Lee[1,2], Colin Boyd[1], Ed Dawson[1],
Kwangjo Kim[3], Jeongmo Yang[2], Seungjae Yoo[2]

[1] Information Security Research Center,
Queensland University of Technology,
GPO Box 2434, Brisbane, QLD 4001, Australia,
{b6.lee,c.boyd,e.dawson}@qut.edu.au
[2] Joongbu University,
101 Daehak-Ro, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea,
{sultan,jmyang,sjyoo}@joongbu.ac.kr
[3] Information and Communications University,
58-4, Hwaam-dong, Yusong-gu, Daejeon, 305-732, Korea,
kkj@icu.ac.kr

Abstract. It had been thought that it is difficult to provide receipt-freeness in mixnet-based electronic voting schemes. Any kind of user chosen randomness can be used to construct a receipt, since a user can prove to a buyer how he had encrypted the ballot. In this paper we propose a simple and efficient method to incorporate receipt-freeness in mixnet-based electronic voting schemes by using the well known re-encryption technique and designated verifier re-encryption proof (DVRP). In our scheme a voter has to prepare his encrypted ballot through a randomization service provided by a tamper resistant randomizer (TRR), in such a way that he finally loses his knowledge on randomness. This method can be used in most mixnet-based electronic voting scheme to provide receipt-freeness.

Keywords. Electronic voting, Receipt-freeness, Mixnet, Re-encryption, Designated-verifier re-encryption proof, Tamper resistant randomizer.

1 Introduction

With the research on electronic voting we try to implement real world voting procedure in electronic means using computer and network. Since we are already accustomed to many electronic means such as high speed computer, Internet, as well as mobile communications, we also expect that voting would be provided through electronic means. It seems to be a worldwide trend that the participation rate in election is decreasing, specially younger generations do not want to participate in voting if they are not interested in. Electronic voting can be a good solution against this problem by providing easier and friendlier means of voting.

Over the world many kind of electoral systems are currently being used [ES]. Those electoral systems can be classified into the following two categories; plurality systems and majoritorian systems. First, the plurality system is relatively simple; the candidate who receives the most votes is elected regardless of whether the candidate receives a majority of votes. Second, the majoritorian system is a little more complicated; a winning candidate is required to receive an absolute majority (more than half) of the vote. In Australian alternative voting (preferential voting) system, voter's next preferences are used to decide a majority winner in later rounds, if one does not emerge from the first round of counting.

In cryptographic research electronic voting is one of the most challenging cryptographic protocol problems, because it has extensive security requirements to be satisfied, and some of them are quite contradictory.

Security requirements.

- **Privacy**: Ensures the secrecy of the contents of ballots. Usually it is achieved by encrypting the ballot with the public key of a group of authorities and by trusting them.
- **Prevention of double voting**: Ensures that any eligible voters can vote only once. To achieve this the authority needs to check the authenticity and eligibility of the voter and record the participation in his secure database. If a public bulletin board is used as a public communication channel (most messages are posted there) and voters access it in an authenticated manner, double voting can be prevented easily.
- **Universal verifiability**: Ensures that any party can be convinced that all valid votes have been included in the final tally. To achieve this all the relevant messages need to be published and the correctness of all processes (voting, mixing, tally) should be publicly verifiable.
- **Fairness**: Ensures that no partial tally is revealed to anyone before the end of the election procedure, as it may affect the voting result in some way. Sometimes we need to trust that the authorities will not reveal partial tally.
- **Robustness**: Ensures that the voting system can tolerate a certain number of faulty participants.
- **Receipt-freeness**: Ensures that a voter neither obtains nor is able to construct a receipt which can prove the content of his vote. To achieve this the voter should not be able to choose his own randomness when he prepares his ballot. A user chosen randomness can work as a receipt.

Sometimes vote buying and coercion issues are discussed as an important security requirement. But, without receipt voters cannot be actively involved in vote buying or coercion. Without receipt these issues would be limited to social problems and are out of the scope of our discussion.

Main Approaches. Electronic voting schemes found in the literature can be classified by their approaches into the following three categories:

- Schemes using blind signature: [Cha88], [FOO92], [Ohk99], [Kim01].

- Schemes using mix-net: [PIK93], [SK95], [Abe98], [Jak98a], [Jak98b], [Abe99], [HS00], [FS01], [Nef01], [Gol02], [Gro03].
- Schemes using homomorphic encryption: [Ben87], [SK94], [CFSY96], [CGS97], [LK00], [HS00], [Hirt01], [Bau01], [Hirt01], [MBC01], [LK02].

Voting schemes based on blind signature are simple, efficient, and flexible, but require the existence of anonymous channel. Frequently, an anonymous channel is implemented using mixnet, but if a secure mixnet is available, a blind signature is not required anymore. Voting schemes based on mixnet are generally not efficient because they require huge amounts of computation for multiple mixers to prove the correctness of their mixing. But recent results of [FS01], [Nef01], [Gro03], and [Gol02] have greatly improved the efficiency of mixnet. Voting schemes based on homomorphic encryption are efficient in the opening stage, but intensive zero-knowledge proofs are used to prove the validity of each ballot in the voting stage, which are costly for the voters. Note that much extensive research on receipt-freeness had been done in this approach.

Our Contribution. In this paper we propose a simple and efficient method to incorporate receipt-freeness in mixnet-based electronic voting schemes by using the well known re-encryption technique and designated-verifier re-encryption proof (DVRP). In our scheme a voter has to prepare his final encrypted ballot through a randomization service provided by a tamper resistant randomizer (TRR), in such a way that he finally loses his knowledge on randomness. This method can be used in most mixnet-based electronic voting schemes to provide receipt-freeness.

Outline of the Paper. The rest of the paper is organized as follows. In Section 2 we review several background concepts and related works in electronic voting. In Section 3 we describe our voting model such as entities and their roles, communication model, and assumptions. In Section 4 we describe some cryptographic primitives such as threshold ElGamal encryption and designated-verifier re-encryption proof, which are required to describe the proposed protocol. The proposed voting protocol is described in Section 5 and is analyzed in Section 6. Finally, we conclude and discuss future works in Section 7.

2 Background and Related Works

2.1 Receipt-freeness

Receipt-freeness is a unique security requirement of electronic voting systems which distinguishes it from other cryptographic protocol problems. Also it is a very important property in electronic voting, since vote buying and coercion are very common experiences in real world election scenarios. It is thought that without providing receipt-freeness electronic voting schemes cannot be used for real political elections.

The concept of receipt-freeness was first introduced by Benaloh and Tuinstra [BT94]. Considering the threat of vote buyers (or coercers), a voting scheme

should ensure not only that a voter *can* keep his vote private, but also that he *must* keep it private. The voter should not be able to prove to a third party that he had cast a particular vote. He must neither obtain nor be able to construct a receipt which can prove the content of his vote.

To achieve receipt-freeness, voting schemes in the literature use some kind of trusted authority which provide a randomization service and make some physical assumption about the communication channel between the voter and the authority depending on the design of the protocol.

1. One-way untappable channel from the voter to the authority [Oka97].
2. One-way untappable channel from the authority to the voter [SK95,HS00].
3. Two-way untappable channel (voting booth) between the voter and the authority [BT94,Hirt01,LK02].

Note that research on receipt-freeness had been done mainly in homomorphic encryption based voting schemes, since designing a receipt-free scheme is relatively easy in those schemes by using zero-knowledge proof techniques. On the other hand, in blind signature based or mixnet-based schemes, voter chosen randomness can be used as a receipt. The voter can prove the content of his encrypted ballot using his knowledge of randomness.

[LK00] tried to provide receipt-freeness by extending [CGS97]. They assumed a trusted third party called honest verifier (HV) who verifies the validity of voter's first ballot and provides a randomization service, *i.e.*, HV re-encrypts it to generate the final ballot and generates the proof of validity of ballot cooperatively with the voter such that the voter cannot obtain any receipt. But [Hirt01] has pointed out that in this protocol a malicious HV can help a voter to cast an invalid vote and thereby falsify the outcome of the whole vote. Moreover the voter can construct a receipt by choosing his challenge as a hash value of his first ballot. This is the same attack applied to [BT94]. To resist against this attack, voter should not be allowed to choose any challenge. [Hirt01] fixed [LK00] and proposed a receipt-free voting scheme based on a third-party randomizer. The role of the randomizer is similar to HV of [LK00], but the randomizer generates the re-encryption proof in a designated-verifier manner and generates the proof of validity using a divertible zero-knowledge proof technique.

[HS00] provided receipt-freeness in homomorphic encryption based voting requiring only one way untappable channel from a randomizer to voters with a cost of huge computation of the randomizer. In this scheme the randomizer works as a kind of personal mixer who presents randomized ballots to a voter in a designated-verifier manner.

[MBC01] proposed a receipt-free electronic voting protocol using a tamper-resistant smartcard which plays the role of personal mixer. But in their voting protocol the re-encryption proof is given in an interactive manner, therefore the same attack applied to [BT94] and [LK00] is possible. [LK02] fixed [MBC01] and proposed a receipt-free electronic voting scheme in which a tamper-resistant randomizer (TRR) replaces the role of untappable channel and a third party randomizer. In this scheme voter prepares an encrypted ballot through an inter-

active protocol with TRR in a way that he loses his randomness but is convinced personally that the final ballot is constructed correctly.

All of these previous works are homomorphic encryption based voting schemes. In this paper we suggest a simple and efficient method to incorporate receipt-freeness in mixnet-based voting schemes by using similar randomization technique. In mixnet-based voting schemes this kind of randomization technique can be applied more efficiently since we do not need to prove the validity of each ballot. To the best of our knowledge, this is the first work to incorporate receipt-freeness in mixnet-based voting schemes.

2.2 Mixnet-based Voting Schemes

The notion of a mixnet was first introduced by Chaum [Cha88], and further developed by a number of researchers. A mixnet enables a set of senders to send their messages anonymously, thus it is a primitive to provide anonymity service. Mixnets can be classified into decryption mixnet and re-encryption mixnet depending on mixing mechanism. The original proposal of Chaum was a decryption mixnet, but many recent works deal with re-encryption mixnet, since it can separate mixing and decryption phases, which provides more flexibility, robustness, and efficiency. Mixnets can also be classified into verifiable mixnet and optimistic mixnet depending on correctness proof.

In verifiable mixnet each server provides proofs that its shuffling is correct, thus the correctness of mixing is publicly verifiable. Abe [Abe99] proposed a general framework of permutation mixnets. Recent works by [FS01], [Nef01], and [Gro03] have shown breakthrough progress in the construction of verifiable mixnet and the proving technique of a correct shuffle. [FS01] represented a shuffling as a matrix between inputs and outputs and proved that it is a correct permutation. [Nef01] has provided verifiable permutation using an iterated logarithmic multiplication proof. [Gro03] used a homomorphic multicommitment scheme to provide a more efficient shuffling proof.

On the other hand, in optimistic mixnet the verification of correct shuffling is not provided by each server. Instead, the correctness of the shuffling of the whole mixnet is verified after the mixnet outputs the shuffling results in plaintexts. Drawbacks of optimistic mixnets include that a cheating server cannot be identified instantly and some outputs are revealed in plaintexts even when the shuffling is incorrect. Jakobsson tried to design efficient optimistic mixnets [Jak98a,Jak98b]. More recently Golle *et. al.* [Gol02] has shown more efficient mixnet by using an optimistic approach, *i.e.*, they proved only the preservation of the product of messages after mixing, not proving the correctness of mixing of each message. But there exists some criticisms [Wik02,AI03] to this approach.

2.3 Tamper-Resistant Hardware Device

To provide receipt-freeness we need to introduce a trusted third party randomizer and untappable channel between voters and the randomizer. But, in the real world, implementing an untappable channel in distributed environments is very

difficult. If a physically isolated voting booth in a dedicated computer network is used to achieve untappable channel, it will be expensive and inconvenient for voters since they have to go to a particular voting booth. Also, assuming a trusted third party randomizer is a burden.

As suggested in [MBC01] and [LK02], a tamper-resistant hardware device can replace the role of untappable channel and a trusted third party. Moreover, a tamper-resistant hardware device is thought to be the ultimate place to store user's secret information such as secret signing key, since it is designed by secure architecture and has limited interface. We expect that it will be available to most users in the near future. In this paper we use a hardware device called tamper resistant randomizer (TRR) to provide a randomization service to voter's encrypted ballot.

3 Voting Model

Overview. A typical mixnet-based electronic voting scheme runs as follows.

1. Voting: A voter prepares an encrypted ballot and posts it on a bulletin board in an authenticated manner with his signature.
2. Mixing: Multiple independent mix servers shuffle the posted ballots sequentially in a verifiable way such that the voter-vote relationship is lost.
3. Tally: After the mixing process is finished, multiple tally servers jointly open encrypted ballots using threshold decryption protocol.

Our main idea is quite simple. We assume a third party randomizer which provides randomization service; it receives the voter's first ballot, randomizes it by using re-encryption to generate a final ballot, and gives it to the voter in an authenticated way. A voter is required to make his final encrypted ballot through an interactive protocol with the randomizer. In this paper the randomizer is implemented by a secure hardware device called tamper resistant randomizer (TRR). In the randomization process, the randomizer provides a designated-verifier re-encryption proof (DVRP) to the voter, so the voter is convinced personally that his final ballot is constructed correctly, but he cannot transfer the proof to others. Through the randomization process the voter loses his knowledge of randomness, thus he cannot construct a receipt.

Entities and Their Roles. Main entities involved in the proposed voting protocol are an administrator A, l voters V_i ($i = \{1, \ldots, l\}$), m mixers M_j ($j = \{1, \ldots, m\}$), and n talliers T_k ($k = \{1, \ldots, n\}$). The roles of each entity are as follows:

- Administrator A manages the whole voting process. A announces the list of candidates, the list of eligible voters, and system parameters including the public key for ballot encryption. A issues TRRs to eligible voters in the registration stage. A publishes the voting result.

- Voter V_i participates in voting. We assume that a voter is certified properly, for example, using PKI, and has a signing key corresponding to the certified public key. V_i needs to identify and register himself to A, then A issues a TRR_i to him which is securely equipped with its own signing key. In the voting stage V_i generates a final encrypted ballot through an interactive protocol with TRR_i and posts it on the bulletin board with his signature.
- Mixers M_j provide mixing service for the collected ballots such that the voter-vote relationship is lost.
- Talliers T_k share the private key of the voting scheme in a (t, n)-threshold verifiable secret sharing (VSS) scheme. After the mixing stage is finished, they cooperatively open each ballot using the (t, n)-threshold decryption protocol.

Tamper resistant randomizer. A tamper resistant randomizer (TRR) is a secure hardware device owned by a voter which works in voter's computer system. It is securely equipped with its own signing key and voter's certified public key. It has the functionality of computing re-encryption and designated-verifier re-encryption proof (DVRP). The communication channel between voter and TRR is an internal channel which does not use network functionality. It is assumed that any party over the network cannot observe the internal communication. It provides a randomization service to voter's encrypted ballot; receives voter's first ballot, re-encrypts it to generate a final ballot, and gives it to the voter in an authenticated manner using its signature. It also provides DVRP to the voter.

Communication Model. In this paper a bulletin board is used as a public communication channel with memory. It can be read by anyone, but only legitimate parties can write messages on it in an authenticated manner. Once a message is written on bulletin board, it cannot be deleted or overwritten. Most messages and proofs of the protocol will be posted on the bulletin board. It is a main communication tool to provide the voting protocol with a universal verifiability. For the voting system to be reliable the bulletin board system should be tamper-proof and resistant against the denial of service (DoS) attack.

The communication channel between involved parties (voters, mixers, talliers) and bulletin board is a public channel such as the Internet. But the communication between a voter and his TRR is an internal channel which cannot be observed by a buyer. We need to assume that a buyer cannot observe the very moment of voter's voting. This is a basic assumption to get receipt-freeness.

Ballot encoding. In the proposed voting protocol any fixed encoding format can be used like most mixnet-based voting schemes, possibly within the size of a modular number. Therefore the proposed protocol provides extensive flexibility and can be used for a wide range of complicated real world voting schemes, for example, Australian preferential voting. Note that, if the encoding format is not fixed, there is a possibility that a specific encoding agreed between a voter and a buyer can be used as a receipt.

4 Cryptographic Primitives

In this paper ballots are encrypted with ElGamal encryption which will be decrypted through a (t, n)-threshold decryption protocol and re-encryption mixnet is used to provide anonymity service.

Consider the ElGamal encryption scheme [ElG85] under a multiplicative subgroup Z_p^* of order q, where p and q are large primes such that $q \mid p-1$. If a receiver chooses a private key s, the corresponding public key is $h = g^s$ where g is the generator of the subgroup. Given a message $m \in \langle g \rangle$, encryption of m is given by $(x, y) = (g^\alpha, h^\alpha m)$ for a randomly chosen $\alpha \in_R Z_q$. To decrypt the ciphertext (x, y), the receiver recovers the plaintext as $m = y/x^s$ using the private key s.

4.1 Re-encryption and Mixnet

ElGamal is a probabilistic encryption scheme that allows re-randomization of ciphertexts. Given an ElGamal ciphertext (x, y), a mix server can efficiently compute a new ciphertext $(x', y') = (xg^r, yh^r)$, choosing $r \in_R Z_q^*$ at random, that decrypts to the same plaintext as (x, y). This re-encryption can be computed by anyone (it does not require the knowledge of the private key) and the exponentiations can be pre-computed.

Given two ElGamal ciphertexts, it is infeasible to determine whether one is a re-encryption of the other without knowledge of either the private key s or the re-encryption factor r, assuming that the Decisional Diffie-Hellman problem is hard in Z_q. Using this property a mix server can hide the correspondence between its input and output ciphertexts, while preserving messages, by outputing re-encrypted ciphertexts in a random order.

4.2 Designated-Verifier Re-encryption Proofs

A designated-verifier proof is a proof which is convincing only to the designated verifier, but it is completely useless when transferred to any other entity [JSI96].

The basic idea is to prove the knowledge of either the witness in question or the secret key of the designated verifier. Such a proof convinces the designated verifier personally because he assumes that the prover does not know his secret key. But, if the proof is transferred to another entity, it loses its persuasiveness completely.

We consider designated-verifier re-encryption proofs (DVRP). Let $(x, y) = (g^\alpha, h^\alpha m)$ be an original ElGamal ciphertext of some message m with a public key $h = g^s$. Let $(x_f, y_f) = (xg^\beta, yh^\beta)$ be a re-encrypted ElGamal ciphertext generated by the prover P (TRR). Let $h_V = g^{s_V}$ be the public key of the verifier V (Voter) corresponding to the private key s_V. P wants to prove to V that his re-encryption was generated correctly in a way that his proof cannot be transferred to others. He will prove that x_f/x and y_f/y have the same discrete logarithm β under bases g and h, respectively.

Designated-verifier re-encryption proof (DVRP):

Prover (TRR):

1. Chooses $k, r, t \in_R Z_q$.
2. Computes $(a, b) = (g^k, h^k)$ and $d = g^r h_V^t$.
3. Computes $c = H(a, b, d, x_f, y_f)$ and $u = k - \beta(c + r)$.
4. Sends (c, r, t, u) to V.

Verifier (Voter):

1. Verifies $c \stackrel{?}{=} H(g^u(x_f/x)^{c+r}, h^u(y_f/y)^{c+r}, g^r h_V^t, x_f, y_f)$.

In this protocol $d = g^r h_V^t$ is a trapdoor commitment (or chameleon commitment) for r and t. Because V knows his private key s_V, he can open d to arbitrary values r' and t' such that $r' + s_V t' = r + s_V t$ holds. V can generate the re-encryption proof for any $(\tilde{x}, \tilde{y})$ of his choice using his knowledge of s_V. Selecting $(\tilde{\gamma}, \tilde{\delta}, \tilde{u})$ at random, V computes

$$\tilde{c} = H(g^{\tilde{u}}(x_f/\tilde{x})^{\tilde{\gamma}}, h^{\tilde{u}}(y_f/\tilde{y})^{\tilde{\gamma}}, g^{\tilde{\delta}}, x_f, y_f),$$

and also computes $\tilde{r} = \tilde{\gamma} - \tilde{c}$ and $\tilde{t} = (\tilde{\delta} - \tilde{r})/s_V$. Then $(\tilde{c}, \tilde{r}, \tilde{t}, \tilde{u})$ is an accepting proof. Therefore designated-verifier re-encryption proof cannot be transferred to others.

4.3 Threshold ElGamal Encryption

A threshold public-key encryption scheme is used to share a secret key among n talliers such that messages can be decrypted only when a substantial subset of talliers cooperate. More detailed description is found in [CGS97] and [Ped91]. It consists of key generation protocol, encryption algorithm, and decryption protocol.

Consider a (t, n)-threshold decryption scheme where the secret key is shared among n talliers T_k $(1 \leq k \leq n)$ and decryption is possible only when more than t talliers cooperate. Through the key generation protocol, each tallier T_k will possess a share $s_k \in Z_q$ of a secret s. Each tallier publishes the value $h_k = g^{s_k}$ as a commitment of the share s_k. The shares s_k are chosen such that the secret s can be reconstructed from any subset Λ of t shares using the appropriate Lagrange coefficients,

$$s = \sum_{k \in \Lambda} s_k \lambda_{k,\Lambda}, \quad \lambda_{k,\Lambda} = \prod_{l \in \Lambda \setminus \{k\}} \frac{l}{l - k}.$$

The public key $h = g^s$ is published to all participants in the system.

Encryption of a message m using the public key h is given by $(x, y) = (g^\alpha, h^\alpha m)$ which is the same as the ordinary ElGamal encryption. To decrypt a ciphertext $(x, y) = (g^\alpha, h^\alpha m)$ without reconstructing the secret s, talliers execute the following protocol:

1. Each tallier T_k broadcasts $w_k = x^{s_k}$ and proves the equality of the following discrete logs in zero-knowledge using the proof of knowledge protocol,

$$\log_g h_k = \log_x w_k.$$

2. Let Λ denote any subset of talliers who passed the zero-knowledge proof. Then the plaintext can be recovered as

$$m = y / \prod_{k \in \Lambda} w_k^{\lambda_{k,\Lambda}}.$$

5 Proposed Voting Protocol

The proposed voting protocol is a good combination of a typical mixnet voting protocol and the randomization technique introduced above. It consists of the following 5 stages.

Stage 1. System setup

Administrator A prepares system parameters of the threshold ElGamal encryption scheme. n talliers T_k jointly execute the key generation protocol of the (t, n)-threshold verifiable secret sharing scheme and publish the public key. Let $h = g^s$ be the public key and s be the private key shared by n talliers. A publishes the list of candidates and other required information on voting.

Stage 2. Registration

Voter V_i identifies and registers himself to A. A checks V_i's eligibility and issues TRR_i which is securely equipped with its own signing key and voter's public key. After the registration deadline has passed, A publishes the list of qualified voters with the certificates of voters and TRRs.

Stage 3. Voting

In the voting stage the voter V_i and his TRR_i compute the final encrypted ballot through the following interactive protocol.

- V_i prepares a ballot message m_i, chooses a random number $\alpha \in_R Z_q^*$, and computes a first ballot as $(x, y) = (g^\alpha, h^\alpha m_i)$. He sends (x, y) to TRR_i.
- TRR_i randomizes (x, y) to generate a final ballot $(x_f, y_f) = (xg^\beta, yh^\beta)$, where β is TRR's secure randomness, and signs it $Sig_{TRR_i}(x_f, y_f)$. It also computes a DVRP as described in Section 4. It computes $(a, b) = (g^k, h^k)$ and $d = g^r h_V^t$, where $k, r, t \in_R Z_q$, and computes $c = H(a, b, d, x_f, y_f)$ and $u = k - \beta(c + r)$. Then (c, r, t, u) is a DVRP for (x_f, y_f). TRR_i sends $Sig_{TRR_i}(x_f, y_f)$ and (c, r, t, u) to the voter.
- V_i verifies the validity of DVRP (c, r, t, u) by

$$c \stackrel{?}{=} H(g^u (x_f/x)^{c+r}, h^u (y_f/y)^{c+r}, g^r h_V^t, x_f, y_f).$$

If V_i is convinced that the final ballot is constructed correctly, V_i double signs the final ballot

$$Sig_{V_i}(Sig_{TRR_i}(x_f, y_f))$$

and posts it on the bulletin board.

Stage 4. Mixing

Before the mixing stage, A verifies the double signatures of voters and their TRRs from the posted ballots, and publishes valid ballots on the bulletin board.

Then m mixers M_j shuffle the ballots sequentially and post the shuffled ballots on the bulletin board. In this stage previously proposed verifiable mixnet protocols such as [Abe99], [FS01], [Nef01], and [Gro03] can be used.

Stage 5. Tallying

For the shuffled ballots n talliers jointly decrypt each ballot using the (t, n)-threshold ElGamal decryption protocol to recover the original ballot messages m_i. Finally, A publishes the tally result.

6 Analysis

The proposed voting protocol satisfies all the security requirements proposed in Section 1.

- **Privacy**: The voter-vote relationship is hidden by the mixing service, so the privacy of voter depends on the security of mixnet. If talliers try to open a ballot before mixing, more than t talliers should cooperate. Assuming the honesty of at least $n - t + 1$ talliers, the secrecy of the vote is preserved.
- **Prevention of double voting**: Since the list of eligible voters are published and voters participate in voting in an authenticated manner (using their signature and TRR's signature), double voting is prevented.
- **Universal verifiability**: Since all the messages in each stage (voting, mixing, and tally) are published in the bulletin board and the correctness of processes is publicly verifiable, any observer can verify the validity of the vote result.
- **Fairness**: Assuming the honesty of at least $n - t + 1$ talliers that they will not cooperate to open the ballot before mixing, no partial tally is revealed, and fairness of voting is guaranteed.
- **Robustness**: Partial failure of some voters can be detected and it does not affect the whole voting protocol. Mixing and tally stages are robust against partial failure of the servers.
- **Receipt-freeness**: No voter can construct a receipt from the messages that he had sent or received, since he had lost his randomness. Although he is convinced that the vote message is preserved in the final ballot by DVRP, he cannot transfer the proof to others. Assuming that a buyer cannot observe the very moment of voter's voting and the communication channel between a voter and his TRR is internal, a voter cannot be coerced into casting a particular vote.

In some papers [Gol02,AI03], preventing ballot copying is discussed. In our construction, a voter can try to copy another voter's ballot because of the malleability of ElGamal encryption. If we want to prevent ballot copying, we have to require the voter to prove his knowledge of randomness. In our construction, voter cannot prove anything since he had lost his randomness. However, note that voter also cannot prove that he had copied a specific ballot (cannot construct a receipt).

The proposed method provides most mixnet-based voting protocols with receipt-freeness in a very efficient manner. All the computation required for TRR is 2 offline exponentiations for re-encryption, 4 offline exponentiations for DVRP creation, and 1 signing. Since all the computation in TRR is offline, it can be pre-computed and is suitable to implement in a hardware device which has limited computational power. On the other hand, the computation required for the voter is 6 online exponentiation for DVRP verification and 1 signature verification. Compared with the homomorphic encryption based receipt-free voting schemes, mixnet-based receipt-free voting is more efficient for voters since complex proof of validity of ballot is not needed.

Since tamper-resistant hardware device is the ultimate place to store user's secret information such as digital signing key, we expect that it will be available to many users in the near future and the proposed voting scheme will be quite reasonable.

7 Conclusion

In this paper we proposed a simple and efficient method to incorporate receipt-freeness in mixnet-based electronic voting schemes by using the well known re-encryption technique and DVRP. In our scheme a voter has to prepare his encrypted ballot through a randomization service provided by TRR in such a way that he finally loses his knowledge on the randomness. This method can be used in most mixnet-based electronic voting schemes to provide receipt-freeness in a very efficient manner.

However, we found that the efficient mixnet of Golle *et. al.* [Gol02] cannot be used in this construction. In their scheme double encryption is used to keep the privacy of vote in a bad case and to support backup mixing. But the inner encryption, which is known only to the voter, can work as a receipt. Applying our tool of receipt-freeness for those efficient mixnets is planned as future work.

8 Acknowledgements

We acknowledge the support of the Australian government through ARC Linkage-International fellowship scheme 2003, Grant No: LX0346868.

References

[Abe98] M. Abe, "Universally verifiable mix-net with verification work independent of the number of mix-servers", *Advances in Cryptology – Eurocrypt'98*, LNCS 1403, Springer-Verlag, pp. 437–447, 1998.

[Abe99] M. Abe, "Mix-networks in permutation networks", *Asiacrypt 1999*, LNCS 1716, Springer-Verlag, pp. 258–273, 1999.

[AI03] M. Abe, and H. Imai, "Flaws in some robust optimistic mixnets", *ACISP 2003*, LNCS 2727, Springer-Verlag, pp. 39–50, 2003.

[Ben87] J. Benaloh, "Verifiable secret-ballot elections", PhD thesis, Yale University, Department of Computer Science, New Haven, CT, September 1987.

[Bau01] O. Baudron, P.-A. Fouque, D. Pointcheval, G. Poupard and J. Stern, "Practical multi-candidate election system", *Proc. of the 20th ACM Symposium on Principles of Distributed Computing*, N. Shavit Ed., pp. 274–283, ACM Press, 2001.

[BT94] J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections", *Proc. of 26th Symp. on Theory of Computing (STOC'94)*, pp. 544–553, New York, 1994.

[CFSY96] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung, "Multi-authority secret ballot elections with linear work", *Advances in Cryptology – Eurocrypt'96*, LNCS 1070, Springer-Verlag, pp. 72–83, 1996.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure an optimally efficient multi-authority election schemes", *Advances in Cryptology – Eurocrypt'97*, LNCS 1233, Springer-Verlag, pp. 103–118, 1997.

[Cha88] D. Chaum, "Elections with unconditionally- secret ballots and disruption equivalent to breaking RSA", *Advances in Cryptology – Eurocrypt'88*, LNCS 330, Springer-Verlag, pp. 177–182, 1988.

[ElG85] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. on IT*, Vol.31, No.4, pp. 467–472, 1985.

[ES] "Electoral systems", Administration and Cost of Elections Project, available at http://www.aceproject.org/main/english/es/

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale election", *Advances in Cryptology – Auscrypt'92*, LNCS 718, Springer-Verlag, pp. 244–260, 1992.

[FS01] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle", *Crypto 2001*, LNCS 2139, Springer-Verlag, pp. 368–387, 2001.

[Gol02] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, and A. Juels, "Optimistic mixing for exit-polls", *Asiacrypt 2002*, LNCS 2501, Springer-Verlag, pp. 451–465, 2002.

[Gro03] J. Groth, "A Verifiable Secret Shuffle of Homomorphic Encryptions", Appears in Public Key Cryptography - PKC 2003, LNCS 2567, Springer-Verlag, pp. 145–160, 2003.

[Hirt01] M. Hirt, "Multi-party computation: Efficient protocols, general adversaries, and voting", Ph.D. Thesis, ETH Zurich, Reprint as vol. 3 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-747-2, Hartung-Gorre Verlag, Konstanz, 2001.

[HS00] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption", *Advances in Cryptology - Eurocrypt2000*, LNCS 1807, Springer-Verlag, pp. 539–556, 2000.

[Jak98a] M. Jakobsson, "A practical mix", *Advances in Cryptology – Eurocrypt'98*, LNCS 1403, Springer-Verlag, pp. 449–461, 1998.

[Jak98b] M. Jakobsson, "Flash mixing", Proc. of the 18th ACM Symposium on PODC'98, pp. 83–89, 1998.

[JSI96] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications", *Advances in Cryptology – Eurocrypt'96*, LNCS 1070, Springer-Verlag, pp. 143–154, 1996.

[Kim01] K. Kim, J. Kim, B. Lee, and G. Ahn, "Experimental design of worldwide Internet voting system using PKI", *SSGRR2001*, L'Aquila, Italy, Aug. 6-10, 2001.

[LK00] B. Lee, and K. Kim, "Receipt-free electronic voting through collaboration of voter and honest verifier", *Proceeding of JW-ISC2000*, pp. 101–108, Jan. 25-26, 2000, Okinawa, Japan.

[LK02] B. Lee, and K. Kim, "Receipt-free electronic voting scheme with a tamper-resistant randomizer", *ICISC 2002*, LNCS 2587, Springer-Verlag, pp. 389–406, 2002.

[MBC01] E. Magkos, M. Burmester, V. Chrissikopoulos, "Receipt-freeness in large-scale elections without untappable channels", *1st IFIP Conference on E-Commerce / E-business / E-Government*, Zurich, Octomber 2001, Kluwer Academics Publishers, pp. 683–693, 2001.

[Nef01] A. Neff, "A verifiable secret shuffle and its application to E-voting", *ACM CCS 2001*, ACM Press, pp. 116–125, 2001.

[Oka97] T. Okamoto, "Receipt-free electronic voting schemes for large scale elections", *Proc. of Workshop on Security Protocols'97*, LNCS 1361, Springer-Verlag, pp. 25–35, 1997.

[Ohk99] M. Ohkubo, F. Miura, M. Abe, A. Fujioka and T. Okamoto, "An improvement on a practical secret voting scheme", *Information Security'99*, LNCS 1729, Springer-Verlag, pp. 225–234, 1999.

[Pai99] P. Paillier, "Public-key cryptosystems based on discrete logarithms residues", *Advances in Cryptology - Eurocrypt '99*, LNCS 1592, Springer-Verlag, pp. 223–238, 1999.

[Ped91] T. Pedersen, "A threshold cryptosystem without a trusted party", *Advances in Cryptology - Eurocrypt '91*, LNCS 547, Springer-Verlag, pp. 522–526, 1991.

[PIK93] C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme", *Advances in Cryptology – Eurocrypt'93*, LNCS 765, Springer-Verlag, pp. 248–259, 1994.

[SK94] K. Sako and J. Kilian, "Secure voting using partial compatible homomorphisms", *Advances in Cryptology – Crypto'94*, LNCS 839, Springer-Verlag, pp. 411–424, 1994.

[SK95] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme – a practical solution to the implementation of a voting booth", *Advances in Cryptology – Eurocrypt'95*, LNCS 921, Springer-Verlag, pp. 393–403, 1995.

[Wik02] D. Wikstrom, "How to break, fix, and optimize optimistic mix for exit-polls", SICS Technical Report, T2002:24, available at http://www.sics.se/libindex.html, 2002.