Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer

Byoungcheon Lee[1] and Kwangjo Kim[2]

[1] Joongbu University
San 2-25, Majon-Ri, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea
sultan@joongbu.ac.kr
[2] Information and Communications University
58-4, Hwaam-dong, Yusong-gu, Daejeon, 305-732, Korea
kkj@icu.ac.kr

Abstract. We investigate the receipt-freeness issue of electronic voting protocols. Receipt-freeness means that a voter neither obtains nor is able to construct a receipt proving the content of his vote. [Hirt01] proposed a receipt-free voting scheme by introducing a third-party randomizer and by using divertible zero-knowledge proof of validity and designated-verifier re-encryption proof. This scheme satisfies receipt-freeness under the assumption that the randomizer does not collude with a buyer and two-way untappable channel exists between voters and the randomizer. But untappable channel is hard to implement in real world and will cause inconvenience to voters although it is provided. In this paper we extend [Hirt01] such that a tamper-resistant randomizer (TRR), a secure hardware device such as smart card or Java card, replaces the role of third-party randomizer and untappable channel. Moreover K-out-of-L receipt-free voting is provided in more efficient manner by introducing divertible proof of difference.

Keywords: Electronic voting, receipt-freeness, tamper-resistant randomizer, divertible zero-knowledge proof.

1 Introduction

The research on electronic voting is very important for the progress of democracy. It is expected that in the near future electronic voting will be used more frequently to collect people's opinion for many kind of political and social decisions through cyber space. In cryptographic aspect it is one of the most significant applications of cryptographic protocols.

1.1 Security Requirements and Approaches

Many extensive researches on electronic voting have been conducted and now an extensive list of security requirements for electronic voting is available. Generally we can classify the security requirements of electronic voting into the following two categories [BT94, FOO92, MH96, NR94, LK00]:

P.J. Lee and C.H. Lim (Eds.): ICISC 2002, LNCS 2587, pp. 389–406, 2003.

Basic Requirements

- Privacy: All votes should be kept secret.
- Completeness: All valid votes should be counted correctly.
- Soundness: Any invalid vote should not be counted.
- Unreusability (prevent double voting): No voter can vote twice.
- Eligibility: No one who is not allowed to vote can vote.
- Fairness: Noting can affect the voting.

Extended Requirements

- Robustness: The voting system should be successful regardless of partial failure of the system.
- Universal verifiability: Anyone can verify the fact that the election is fair and the published tally is correctly computed from the ballots that were correctly cast.
- Receipt-freeness: A voter neither obtains nor is able to construct a receipt proving the content of his vote.
- Incoercibility: A voter cannot be coerced into casting a particular vote by a coercer. This is a stronger requirement than receipt-freeness. If we assume that the coercer cannot observe the voter during the very moment of voting, receipt-freeness gives incoercibility and vote buying is prevented.

The basic requirements are satisfied in most electronic voting systems and their implementation is relatively easy. But the extended requirements are hard to implement and in many case they require large amount of computation and communication. Specially universal verifiability and receipt-freeness seem to be contradictory. Exchanged messages or user-chosen randomness are useful to verify the correctness of vote, but there are possibilities that these data are used as a receipt. Current research on electronic voting is focused on receipt-free schemes that also satisfy universal verifiability.

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

- Schemes using blind signature: [Cha88, FOO92, OMAFO99].
- Schemes using mix-net: [PIK93, SK95, Pfi94, MH96, Abe98, Jak98, HS00, Hirt01, MBC01].
- Schemes using homomorphic encryption: [Ben87, SK94, CFSY96, CGS97, LK00, Hirt01, BFPPS01, Cha02, Po00].

Voting schemes based on blind signature technique are simple, efficient, and flexible, but they cannot provide receipt-freeness. Voter's blind factor can be used as a receipt of his vote, therefore a voter can prove his vote to a buyer. Voting schemes based on mix-net are generally not efficient because they require huge amount of computation for multiple mixers (mixing and proving correctness of their jobs). Voting schemes based on homomorphic encryption use zero-knowledge proof techniques to prove the validity of ballot. In this approach there have been extensive researches to provide receipt-freeness.

1.2 Approaches to Achieve Receipt-Freeness

The concept of receipt-freeness was first introduced by Benaloh and Tuinstra [BT94]. Considering the threat of vote-buyers (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 proving the content of his vote. Recently, [HS00] has shown that the voting protocol of [BT94] does not provide receipt-freeness.

In this study we assume that the coercer does not observe the voter during the very moment of voting. Obviously, if voters use personal computer to vote over the Internet, the coercer can manage to observe the voter and coerce him to cast a particular vote. But this threat is possible in any voting system using personal computer over the Internet and is beyond the scope of cryptographic research. Our goal in this paper is to prevent a voter from getting or being able to construct a receipt.

To achieve receipt-freeness, voting schemes in the literature make some physical assumption about the communication channel between the voter and the authority.

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].

Note that the existence of untappable channel from the authority to the voter is the weakest physical assumption for receipt-freeness [HS00].

1.3 Related Works

In this section, we review [LK00, Hirt01], and [MBC01] briefly because our study is based on their results.

[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 generates the final ballot and proof of validity of ballot cooperatively with the voter such that the voter cannot get any receipt. This is an efficient solution because a single entity can provide receipt-freeness. 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] proposed a receipt-free voting scheme based on a third-party randomizer. The role of randomizer is similar to HV of [LK00] (generates the final ballot by randomizing the first ballot and generates the proof of validity interactively with the voter), but the randomizer generates the re-encryption proof in

designated-verifier way and uses a divertible zero-knowledge proof technique to generate the proof of validity. Recently [BFPPS01] proposed an efficient multi-candidate electronic voting scheme based on Paillier Cryptosystem [Pai99], in which tallying stage is more efficient.

[MBC01] proposed a receipt-free electronic voting protocol using a tamper-resistant smartcard. They pointed out the difficulty of implementing untappable channel and introduced the necessity of tamper-resistant device. In their voting protocol smartcard plays the role of mixer. But, in their voting protocol the re-encryption proof is given in an interactive way, so the same attack applied to [BT94] and [LK00] is possible. The re-encryption proof should be given in a non-interactive and designated-verifier way such that it cannot be transferred to third parties and the voter cannot construct a receipt.

1.4 Tamper-Resistant Hardware Device

[HS00] stated that the existence of untappable channel from the authority to the voter is the weakest physical assumption for receipt-freeness. But, in the real world, implementing an untappable channel in distributed environment is very difficult. If a physically isolated voting booth in a dedicated computer network is used to achieve receipt-freeness, it will cost a lot and will cause inconvenience to voters since they have to go to particular voting booth. If the overall voting system is inconvenient, participation in electronic voting will not be advantageous.

To increase the participation rate in electronic voting, Internet voting will be the best solution, in which voters can participate in electronic voting in any place over the Internet. But achieving receipt-freeness is a hard task in Internet voting, since Internet is a tappable channel.

As suggested in [MBC01], a tamper-resistant hardware device can replace the role of untappable channel and trusted third party. Since tamper-resistant hardware devices are designed by secure architecture, it is thought to be the ultimate place to store user's secret information such as secret signing key. As the technology of tamper-resistant hardware device advances in the point of computational power, it can compute complicated computation. Recently, the technology of tamper-resistant hardware device advances quickly and the usage of smart card and Java card is increasing. Therefore tamper-resistant hardware device seems to be more practical assumption than untappable channel and trusted third party. It is expected that tamper-resistant hardware device can be applied to wide range of advanced applications in the near future. Electronic voting can be a good example.

1.5 Our Contribution

In this paper we extend [Hirt01] scheme such that a tamper-resistant randomizer (TRR), a secure hardware device such as smart card or Java card, replaces the role of third party randomizer and untappable channel. Moreover K-out-of-L (choose K candidates among L candidates) receipt-free voting is provided in

more efficient manner by introducing divertible proof of difference. In this scheme TRR is locally connected to the voter system (does not use network facility) and executes the role of randomizer. This scheme does not require untappable communication channel and trusted third party. Assuming the tamper-resistance of TRR, it provides receipt-freeness together with efficiency. Furthermore we consider an efficient variant that the voter just inputs his choice, and then TRR generates encrypted ballot and proof of validity, and finally the voter approves the result.

1.6 Outline of the Paper

The paper is organized as follows. In Section 2, we overview the proposed voting scheme briefly and describe the model of electronic voting. Cryptographic primitives are described in Section 3 and complete voting protocol is described in Section 4. Security and efficiency analysis are followed in Sections 5 and 6. Finally we conclude in Section 7.

2 Model of Electronic Voting

In this section we overview the proposed voting scheme briefly and describe the model of electronic voting. Some of the zero-knowledge proof techniques which appear first in this section will be described in the following section.

2.1 Overview of the Proposed Voting Protocol

The proposed voting protocol runs as follows. The voter generates an encrypted first ballot and gives it to tamper-resistant randomizer (TRR). Then TRR randomizes it to generate a final ballot and prove its correctness to the voter using the designated-verifier re-encryption proof. If this is valid, the voter and TRR jointly generate a proof of validity of the final ballot using divertible proof of validity protocol and divertible proof of difference protocol. The final ballot and the proof of validity are first digitally signed by voter's TRR during the protocol run, and then they are signed by the voter to represent voter's approval. The voter posts the final ballot, the proof of validity and the proof of difference on the bulletin board. Only valid ballots are counted by the authority.

2.2 Entities

The main entities involved in the voting protocol are an administrator A, M voters V_i $(i = 1, \ldots, M)$, and N talliers T_j $(j = 1, \ldots, N)$. To participate in the voting each voter should have his own tamper-resistant randomizer (TRR) issued by A. The roles of each entity are as follows:

- Administrator A verifies the identities and their eligibilities of M voters and then issues TRR devices to voters in the registration stage. She manages the whole voting process (announces the list of candidates, collects valid ballots, and announces the final result).

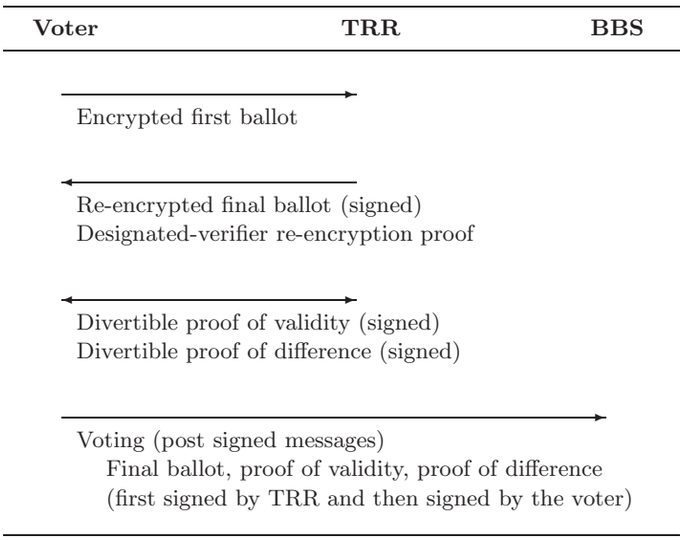

Fig. 1. Overview of the proposed voting protocol

- There are M voters V_i $(i = 1, \ldots, M)$. They have their own digital signature keys certified by a certification authority (CA). To participate in the voting, each voter needs to register to A and get his own TRR issued by A.
- There are N talliers T_j $(j = 1, \ldots, N)$ who cooperatively decrypt the collected ballots to open the result of voting. A threshold t denotes the lower bound of the number of authorities that is guaranteed to remain honest during the protocol.

Here we assume that the administrator A does not collude with a buyer to issue an illegal TRR to a voter. This assumption is equivalent to the assumption of [Hirt01] that the third party randomizer does not collude with a buyer.

2.3 Tamper-Resistant Randomizer

TRR is a tamper-resistant hardware device issued by the administrator A (or any trusted third party) to a specific qualified voter. It is not an independent entity in our model, but is a hardware equipment owned by the voter. It is directly connected to voter system and has restricted set of interfaces for communication. The communication channel between the voter and his TRR is assumed to be untappable.

It has its own randomness source and is securely equipped with its own digital signature key certified by the administrator. It is equipped with talliers' public key and voter's public key. Because it is a tamper-resistant device, even the administrator and the voter cannot access the randomness and any internal information.

It helps the voter to generate an encrypted ballot and proof of validity such that the voter can be convinced of the validity of his vote but cannot get a receipt of his vote. More specifically, TRR produces the final ballot by randomizing voter's first ballot, provides designated-verifier re-encryption proof, produces proof of validity jointly with the voter. All messages that TRR provides are digitally signed with its signature key. Only the encrypted ballots and the proof of validity which are signed by TRR are accepted to be valid.

2.4 Communication Model

The communication channel between the voter and the administrator is a public broadcast channel with memory, *i.e.*, a bulletin board. Voters post their encrypted ballot and proof of validity on the bulletin board with their signature, so double voting is prevented. Anyone except the voter cannot post a ballot with the name of the voter. Anyone can read and verify the posted ballots, which provides universal verifiability.

The communication channel between the voter and his TRR is an internal communication without using network functions. We assume that the coercer does not observe the voter during the very moment of voting. Obviously, if voters use personal computer to vote over the Internet, the coercer can manage to observe the voter. But this threat is possible in any voting system using personal computer over the Internet and is beyond the scope of cryptographic research. Our goal in this paper is to prevent a voter from getting or being able to construct a receipt.

2.5 Encoding of Ballots

First, we consider a 1-out-of-L voting scheme in which voters choose a candidate out of L candidates. Let g be a generator of a multiplicative subgroup Z_p^* of order q and h be the public key of talliers. To achieve simple decryption using the homomorphic property of ElGamal encryption, a vote for the i-th candidate $(1 \leq i \leq L)$ is represented as $g^{M^{i-1}}$ where M is the maximum number of voters. Then ElGamal encryption for the vote is given by $(x, y) = (g^\alpha, h^\alpha g^{M^{i-1}})$ where α is voter's random number. This encoding allows easy decoding of the sum by simple remaindering.

Next, we consider a K-out-of-L voting scheme in which voters can have K choices out of L candidates. In this case the total ballot is composed of K independent ballots of 1-out-of-L voting with additional proofs that the K choices are all different.

3 Cryptographic Primitives

3.1 Threshold ElGamal Encryption

To generate encrypted ballot, homomorphic ElGamal encryption and threshold ElGamal decryption are used.

Consider the ElGamal encryption system [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 $m \in Z_p$, 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.

In our proposed voting scheme, we consider a K-out-of-L voting where K is the number of voter's choices and L is the number of candidates. We implement it as K independent ballots of 1-out-of-L voting. If we choose a special encoding of message such that the homomorphic property is preserved, the final tally can be computed by a single decryption of the product of all valid ballots.

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

$$s = \sum_{j \in \Lambda} s_j \lambda_{j,\Lambda}, \quad \lambda_{j,\Lambda} = \prod_{l \in \Lambda \setminus \{j\}} \frac{l}{l-j}$$

The public key $h = g^s$ is announced 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_j broadcasts $w_j = x^{s_j}$ and proves the equality of the following discrete logs in zero-knowledge using the proof of knowledge protocol.

$$\log_g h_j = \log_x w_j.$$

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

$$m = y / \prod_{j \in \Lambda} w_j^{\lambda_{j,\Lambda}}.$$

3.2 Designated-Verifier Re-encryption Proofs

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

The basic idea is to prove knowledge of either the witness in question or of the secret key of the designated verifier. Such a proof convinces the designated verifier 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. Let $(x, y) = (g^l, h^l m)$ be an original ElGamal ciphertext of some message m with a public key $h = g^s$. Let $(x_f, y_f) = (xg^w, yh^w)$ 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 same discrete logarithm under bases g and h, respectively.

Designated-Verifier Re-encryption Proof :

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 $u = k - w(c + r)$.
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 $(\alpha, \beta, \tilde{u})$ at random, V computes

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

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

3.3 Divertible Proof of Validity

In the proposed receipt-free voting scheme, the voter gives his first encrypted ballot to TRR, then TRR re-encrypts it to generate the final ballot. The divertible proof of validity is an interactive modification of the non-interactive proof of validity of ballot such that TRR adds its own randomness to the commitment of the voter and then adjusts the response of the voter such that the non-interactive proof of validity holds for the final ballot, but the voter cannot construct any receipt.

Let $(x, y) = (g^{\alpha}, h^{\alpha} m_i)$ be voter's first ballot for his vote m_i where α is voter's random number and $(x_f, y_f) = (xg^{\beta}, yh^{\beta})$ be the final ballot re-encrypted by TRR where β is TRR's internal random number. Voter and TRR can jointly compute a non-interactive proof of validity for the final ballot as follows:

Divertible Proof of Validity:

1. Voter → TRR (commitment):
   - Voter chooses a random number $w \in_R Z_q$ and computes $a'_i = g^w, b'_i = h^w$.
   - For $j = 1, \ldots, i-1, i+1, \ldots, L$, voter chooses $r'_j, d'_j \in_R Z_q$, and computes $a'_j = g^{r'_j} x^{d'_j}$ and $b'_j = h^{r'_j}(y/m_j)^{d'_j}$.
   - Voter sends $(A', B') = (a'_1, b'_1, \ldots, a'_L, b'_L)$ to TRR.
2. Voter ← TRR (randomized commitment):
   - For $j = 1, \ldots, L$, TRR chooses $r''_j, d''_j \in_R Z_q$, and computes $a_j = a'_j g^{r''_j} x^{d''_j}$ and $b_j = b'_j h^{r''_j}(y/m_j)^{d''_j}$. Here $\sum_j d''_j = 0$ should hold.
   - TRR sends $(A, B) = (a_1, b_1, \ldots, a_L, b_L)$ to the voter.
3. Voter → TRR (response):
   - Voter computes $c = H(a_1, b_1, \ldots, a_L, b_L)$.
   - Voter computes $d'_i = c - \sum_{j \neq i} d'_j$ and $r'_i = w - \alpha d'_i$.
   - Voter sends $(D', R') = (d'_1, r'_1, \ldots, d'_L, r'_L)$ to TRR.
4. Voter ← TRR (adjusted response):
   - For $j = 1, \ldots, L$, TRR computes $d_j = d'_j + d''_j$ and $r_j = r'_j + r''_j - d_j\beta$.
   - TRR sends $(D, R) = (d_1, r_1, \ldots, d_L, r_L)$ to the voter.
5. Voter (Any verifier):
   - Voter checks

$$d_1 + \cdots + d_L \stackrel{?}{=} H(g^{r_1} x_f^{d_1}, h^{r_1}(y_f/m_1)^{d_1}, \ldots, g^{r_L} x_f^{d_L}, h^{r_L}(y_f/m_L)^{d_L}).$$

The final verification equation holds because of the following relations.

$$\begin{aligned} c &= \sum_j d_j \\ a_j &= a'_j g^{r''_j} x^{d''_j} = g^{r'_j + r''_j} x^{d'_j + d''_j} = g^{r_j + \beta d_j} x^{d_j} = g^{r_j} x_f^{d_j}, \\ b_j &= b'_j h^{r''_j}(y/m_j)^{d''_j} = h^{r'_j + r''_j}(y/m_j)^{d'_j + d''_j} = h^{r_j + \beta d_j}(y/m_j)^{d_j} \\ &= h^{r_j}(y_f/m_j)^{d_j}. \end{aligned}$$

Through an interactive protocol between the voter and TRR, voter gets a proof of validity (A, B, D, R) for the final ballot (x_f, y_f). In this protocol, protocol messages from TRR should be authentic, *i.e.*, messages (A, B) and (D, R) should be digitally signed by TRR's private key and verified by the voter. Signed proofs represent that they are generated by TRR.

The original interactive proof of validity protocol is honest-verifier zero-knowledge, *i.e.*, it is zero-knowledge with an honest verifier who selects the challenge independently from the commitment message. The non-interactive variant of proof of validity is zero-knowledge in the random oracle model since the hash value of commitment message is used as a challenge. Since the modified commitment and adjusted response are fully randomized by TRR, the voter cannot prove any correspondence between the proof of validity of the final ballot and that of his first ballot. Therefore this protocol is receipt-free.

3.4 Divertible Proof of Difference

When the voter participates in a K-out-of-L voting, he prepares K independent encrypted ballots and provides proofs that they are all different. Using the same method, the proof of difference can be made divertible and receipt-free.

Let (x_1, y_1) and (x_2, y_2) be two independent first ballots of the voter and (x_{f1}, y_{f1}) and (x_{f2}, y_{f2}) be corresponding final ballots re-encrypted by TRR.

$$(x_1, y_1) = (g^{\alpha_1}, h^{\alpha_1} m_1), (x_2, y_2) = (g^{\alpha_2}, h^{\alpha_2} m_2).$$

$$(x_{f1}, y_{f1}) = (x_1 g^{\beta_1}, y_1 h^{\beta_1}), (x_{f2}, y_{f2}) = (x_2 g^{\beta_2}, y_2 h^{\beta_2}).$$

Now consider their differences as follows.

$$(x, y) \equiv (x_1/x_2, y_1/y_2) = (g^{\alpha_1 - \alpha_2}, h^{\alpha_1 - \alpha_2} m_1/m_2) \equiv (g^{\alpha}, h^{\alpha} m_1/m_2)$$

$$(x_f, y_f) \equiv (x_{f1}/x_{f2}, y_{f1}/y_{f2}) = (x g^{\beta_1 - \beta_2}, y h^{\beta_1 - \beta_2}) \equiv (x g^{\beta}, y h^{\beta})$$

Voter and TRR jointly generate the proof of difference as follows.

Divertible Proof of Difference:

1. Voter $\rightarrow$ TRR (commitment):
   - Voter chooses random numbers $k'_1, k'_2 \in_R Z_q$ and computes $a'_1 = g^{k'_1}, b'_1 = h^{k'_1}, a'_2 = g^{k'_2}, b'_2 = h^{k'_2}$.
   - Voter sends (a'_1, b'_1, a'_2, b'_2) to TRR.
2. Voter $\leftarrow$ TRR (randomized commitment):
   - TRR chooses random numbers $k_1, k_2 \in_R Z_q$ and computes $a_1 = a'_1 g^{k_1}, b_1 = b'_1 h^{k_1}, a_2 = a'_2 g^{k_2}, b_2 = b'_2 h^{k_2}$.
   - TRR sends (a_1, b_1, a_2, b_2) to the voter.
3. Voter $\rightarrow$ TRR (response):
   - Voter computes $c = H(a_1, b_1, a_2, b_2)$.
   - Voter computes $s'_1 = k'_1 - c\alpha, s'_2 = k'_2 - ck'_1$.
   - Voter sends (s'_1, s'_2) to TRR.
4. Voter $\leftarrow$ TRR (adjusted response):
   - TRR computes $c = H(a_1, b_1, a_2, b_2)$.
   - TRR computes $s_1 = s'_1 + k_1 - c\beta = k_1 + k'_1 - c(\alpha + \beta)$ and $s_2 = s'_2 + k_2 - ck_1 = k_2 + k'_2 - c(k_1 + k'_1)$.
   - TRR sends (s_1, s_2) to the voter.
5. Voter (Any verifier):
   - Voter verifies the validity of proof as $a_1 \stackrel{?}{=} g^{s_1} x_f^c$, $a_2 \stackrel{?}{=} g^{s_2} a_1^c$, $b_2 \stackrel{?}{=} h^{s_2} b_1^c$
   - Voter verifies the difference $b_1 \stackrel{?}{=} h^{s_1} y_f^c$.
     If they are equal, it means that two final ballots (x_{f1}, y_{f1}) and (x_{f2}, y_{f2}) are votes for the same candidate, therefore they are not valid. If they are not equal, two final ballots are valid.

The final verification equations hold because of the following relations.

$$a_1 = g^{k_1+k_1'} = g^{s_1+c(\alpha+\beta)} = g^{s_1} x_f^c$$
$$a_2 = g^{k_2+k_2'} = g^{s_2+c(k_1+k_1')} = g^{s_2} a_1^c$$
$$b_2 = h^{k_2+k_2'} = h^{s_2+c(k_1+k_1')} = h^{s_2} b_1^c$$
$$b_1 = h^{k_1+k_1'} = h^{s_1+c(\alpha+\beta)} = h^{s_1} y_f^c.$$

Through an interactive protocol between the voter and TRR, voter gets a proof of difference $(a_1, b_1, a_2, b_2, s_1, s_2)$ for two final ballots (x_{f1}, y_{f1}) and (x_{f2}, y_{f2}). In this protocol, protocol messages from TRR should be authentic, *i.e.*, messages (a_1, b_1, a_2, b_2) and (s_1, s_2) should be digitally signed by TRR's private key and verified by the voter.

Similarly this protocol is zero-knowledge in the random oracle model and is receipt-free.

4 Proposed Receipt-Free Electronic Voting Scheme

The proposed receipt-free electronic voting scheme consists of the following 4 stages: system set-up, registration, voting, and tallying.

Stage 1. System Set-Up
N talliers $(T_1, \ldots, T_N)$ execute the key generation protocol of (t, N)-threshold ElGamal encryption scheme and as a result each tallier T_i possesses his share $s_i \in Z_q$ of a secret s. The resulting public key of the voting system $h = g^s$ is announced to voters. Any cooperation of more than t talliers can decrypt an encrypted ballot. The administrator A publishes the list of L candidates on the bulletin board.

Stage 2. Registration
We assume that every voters V_i have their certificates $Cert_i$ certified by a certification authority (CA). Voter V_i connects to A and requests registration for voting with his certificate, then A verifies V_i's identity and qualification for voting. If V_i is a legitimate voter, A issues a tamper-resistant randomizer TRR_i to V_i in which a digital signature key is equipped securely, and also issues a certificate $CertTRR_i$ which corresponds to TRR_i's digital signature key. In TRR_i, talliers' public key h and voter's certificate $Cert_i$ are equipped. A publishes $(V_i, Cert_i, CertTRR_i)$ on the bulletin board.

Stage 3. Voting
In this stage voter V_i and his TRR_i jointly generates encrypted ballots and proofs of validity as follows. First we consider the 1-out-of-L voting scheme.

1. V_i chooses a candidate among L candidates. Let's assume that he has chosen j-th candidate. He computes his first ballots as $(x, y) = (g^\alpha, h^\alpha g^{M^{j-1}})$ where α is V_i's random number. He sends it to TRR_i with his signature.

2. TRR_i verifies V_i's signature in his first ballot and computes the final ballot as $(x_f, y_f) = (xg^\beta, yh^\beta)$ where β is TRR_i's random number. It also computes the designated-verifier re-encryption proof. It digitally signs the final ballot and the designated-verifier re-encryption proof and sends them to V_i.
3. V_i verifies the digital signature of the final ballot and also verifies its correctness with the designated-verifier re-encryption proof.
4. If the final ballot is generated correctly, V_i and TRR_i jointly compute the proof of validity of the final ballot using the divertible proof of validity protocol. As a result of this protocol, V_i gets the proof of validity, (A, B) and (D, R), which are digitally signed by TRR_i.
5. V_i signs the final ballots and the proof of validity with his private key corresponding to his certificate $Cert_i$, and posts these messages on the bulletin board.

Therefore the posted messages $(x_f, y_f), (A, B), (D, R)$ are first signed by TRR_i and then signed by V_i. Anyone can verify the fact that these messages are generated by TRR_i and approved by V_i.

In the case of K-out-of-L voting scheme, V_i and TRR_i compute K independent final ballots and proofs of validity in the same way. In addition, V_i and TRR_i compute $K-1$ proofs of difference using the divertible proof of difference protocol, which represents that K final ballots are votes for different candidates.

Stage 4. Tallying

When the deadline of voting is reached, administrator A collects all the valid ballots, computes the product $(X, Y) = (\prod_{i=1}^{l} x_{f,i}, \prod_{i=1}^{l} y_{f,i})$ where l is the total number of valid ballots, and posts it on the bulletin board. Anybody can check the validity of the product because all the final ballots are posted on the bulletin board and their validity can be verified publicly. Then N talliers jointly execute the (t, N)-threshold decryption protocol for (X, Y) to obtain $W = Y/X^s$. Because the secret key s is shared among N talliers, any subset of t talliers can decrypt (X, Y) to obtain W. Note that the secret key s is not reconstructed but just X^s is computed in the decryption process.

Now we get $W = g^{r_1 M^0 + r_2 M^1 + \cdots + r_L M^{L-1}}$ where $(r_1, \ldots, r_L)$ are the result of the election. Computation of $(r_1, \ldots, r_L)$ requires the computation of the discrete logarithm problem and it is generally considered as a computationally hard problem. In this case, it requires $\mathcal{O}(\sqrt{l}^{L-1})$ time to get the result [CGS97]. It is feasible only for a reasonable size of l and L. Therefore, if this scheme is applied to a large scale electronic voting, A can group the valid ballots into several subgroups with reasonable size of l, and then N talliers can decrypt the subproducts easily, one by one. Note that this kind of local tallying is a common experience in the real world.

Now we consider two simple variants of the proposed voting protocol.

Non-interactive Variant: If we assume that TRR is tamper-resistant and is constructed correctly by the administrator A, then the first ballot needs not be

encrypted by the voter. In this case, we can consider a variant of the voting protocol that the voter just sends his choices to TRR and then TRR computes by itself (non-interactively) the final encrypted ballots, designated-verifier encryption proofs, proofs of validity, and proofs of difference, with its digital signature. After receiving the results from TRR, the voter approves the results with his digital signature and then posts them. Then the ballot generation protocol can be executed in a non-interactive way and the overall voting protocol will be much more efficient.

Multiple-Choice Variant: Another simple variant is that the proposed scheme can be used to allow duplicated selection of the same candidate, if the proof of difference is not used. In this case the voter can choose K choices out of L candidates without any requirement for difference.

5 Security Analysis

The proposed electronic voting protocol satisfies the basic and extended requirements of electronic voting.

- Privacy: The tallying procedure is executed only for the product of multiple valid ballots. Assuming the honesty of at least $N - t$ talliers (do not open single voter's ballot), privacy of individual voter is satisfied. Since the proof of validity is zero-knowledge, no partial information on voter's choice is exposed.
- Completeness: The final ballot and the proof of validity are posted on the public bulletin board. Anyone can verify the validity of the final ballots, the correctness of ballot collection and the final result. Therefore valid ballots are counted correctly.
- Soundness: Any invalid ballot is detected from the public bulletin board, so it cannot be counted.
- Unreusability (prevent double voting): Each voter posts his encrypted ballot and proofs on the bulletin board with his signature and TRR's signature. Therefore he can vote only once and double voting is detected easily.
- Eligibility: Legitimate voters registered to the administrator A are published on the bulletin board together with their certificates. Therefore only legitimate voters can participate in voting.
- Fairness: Because the privacy of voter is kept by N talliers and the voting protocol is zero-knowledge, nothing can affect the voting process.
- Robustness: (t, N) threshold ElGamal encryption scheme can tolerate the failure of maximum $N - t$ talliers.
- Universal verifiability: Because the final ballot and proof messages are posted on the bulletin board together with voter information, the validity of each ballot is publicly verifiable. The product of valid ballots and tallying result are also publicly verifiable.

- Receipt-freeness: Since the designated-verifier re-encryption proof given by TRR cannot be transferred to others, the voter cannot prove any relation between his first ballot and the final ballot. Since the proof of validity and the proof of difference are fully randomized by TRR, these proof messages are independent from voter's commitment messages. Therefore the voter cannot prove any correlation between the proof messages and his first ballot. Assuming the tamper-resistance of TRR, the voter cannot obtain any information on TRR's internal randomness. Therefore the voter cannot construct any receipt from the protocol messages.
- Incoercibility: Since we have assumed that the coercer cannot observe the voter during the very moment of voting, receipt is the only way for the coercer to check voter's vote. Since the proposed voting scheme satisfies receipt-freeness, incoercibility is also satisfied and vote buying is prevented.

6 Efficiency Analysis

Let's consider the message size transferred in the voting stage and the number of modular exponentiations in the voting stage. Let $|p|$ be the bit size of group element in Z_p, $|q|$ be the bit size of Z_q, and $|s|$ be the bit size of digital signature. In the K-out-of-L voting scheme, exchanged messages are as follows.

- $(2LK + 6K - 4)|p| + (2LK + 2K - 2)|q| + 5|s|$ (from the voter to TRR).
- $(2LK + 6K - 4)|p| + (2LK + 6K - 2)|q| + 5|s|$ (from TRR to the voter).
- $(2LK + 6K - 4)|p| + (2LK + 2K - 2)|q| + 6|s|$ (posted on the bulletin board).

On the other hand the total number of modular exponentiations are given as follows, excluding the digital signature operations.

- Exponentiations by the voter: $8LK + 18K - 12$.
- Exponentiations by TRR: $4LK + 10K - 4$.

Therefore overall performance requires $\mathcal{O}(LK)$ message transfer and modular exponentiations. This is much more efficient compared with [Hirt01] which requires $\mathcal{O}({}_L C_K) \approx \mathcal{O}(2^L)$ message transfer and modular exponentiations. [Hirt01] also introduced a variant using a binary encoding of ballot and a proof of summation which requires $\mathcal{O}(2L)$ message transfer and modular exponentiations.

In this scheme valid ballot and its proof of validity are generated only in the voter system without any network communication. Therefore this scheme is more efficient than [Hirt01] in the point of network communication. The non-interactive variant of the proposed scheme is more simple and efficient in the sense that the inner communication protocol between the voter and TRR is also non-interactive.

The usage of TRR can be considered to be very costly in large scale election. But it is much more practical than the untappable channel assumption. Moreover tamper-resistant hardware devices are thought to be the ultimate place to store

user's secret information, such as secret signing key. As the technology of tamper-resistant hardware device advances in the point of computational power and cost, it is expected that in the near future everybody can store their signing key in their ID card. If this is the case, the proposed electronic voting scheme can be applied very easily over the public network like the Internet without any extra cost.

7 Conclusion

In this paper we have proposed an efficient receipt-free electronic voting scheme using TRR. Because TRR is locally connected to voter system and any network communication is not used during the voting stage, untappable channel assumption is not required and the voting scheme is much more secure and efficient. TRR can be considered to be a secure implementation of the untappable channel and the trusted third party.

For an efficient implementation of K-out-of-L voting, we have extended [Hirt01] using the divertible proof of difference. Our scheme requires $\mathcal{O}(LK)$ message transfer and modular exponentiations while [Hirt01] requires $\mathcal{O}({}_LC_K) \approx \mathcal{O}(2^L)$.

Furthermore we have considered a non-interactive variant that the voter just sends his choices to TRR and then TRR computes by itself the final encrypted ballots, designated-verifier encryption proofs, proofs of validity, and proofs of difference, with its digital signature. Finally, the voter approves the results with his digital signature and then posts them. Then the ballot generation protocol can be executed in non-interactive way and the overall voting protocol can be much more efficient.

Because of the rapid advance of hardware technology, tamper-resistant hardware device tends to have more powerful computation and communication functionality. Moreover it is considered to be the ultimate place to store user's secret information, such as secret signing key. It is expected that it can be applied to wide range of advanced applications in the near future. Therefore TRR seems to be a more practical assumption than untappable channel and trusted third party.

If we can use the Internet for electronic voting, voters can participate in voting in any place they like over the Internet. Then electronic voting system can play an important role to increase the participation rate in voting and realize participatory democracy.

Acknowledgements

We would like to thank many anonymous reviewers for their valuable comments, which help to make this paper more readable one. There was a comment (and we agree) that the proof of difference may leak some information of voter's vote, more than just the fact of difference. Further works need to be done to design more efficient ballot encoding and to improve the proof of difference.

References

[Abe98] M. Abe, "Universally verifiable mix-net with verification work independent of the number of mix-servers", *Advances in Cryptology – Eurocrypt'98*, LNCS Vol.1403, pages 437–447, Springer-Verlag, 1998. 390

[Ben87] J. Benaloh, "Verifiable secret-ballot elections", PhD Thesis, Yale University, Department of Computer Science, New Haven, September 1987. 390

[BFPPS01] 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, Pages 274–283, ACM Press, 2001. 390, 392

[BT94] J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections", *Proc. of 26th Symp. on Theory of Computing (STOC'94)*, pages 544–553, New York, 1994. 389, 391, 392

[CFSY96] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung, "Multi-authority secret ballot elections with linear work", *Advances in Cryptology – Eurocrypt'96*, LNCS Vol.1070, pages 72–83, Springer-Verlag, 1996. 390

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure an optimally efficient multi-authority election schemes", *Advances in Cryptology – Eurocrypt'97*, LNCS Vol.1233, pages 103–118, Springer-Verlag, 1997. 390, 391, 401

[Cha88] D. Chaum, "Elections with unconditionally- secret ballots and disruption equivalent to breaking RSA", *Advances in Cryptology – Eurocrypt'88*, LNCS Vol.330, pages 177–182, Springer-Verlag, 1988. 390

[Cha02] D. Chaum, "Privacy Technology: A survey of security without identification", IACR Distinguished Lecture in Crypto2002, 2002. 390

[ElG85] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. on IT*, Vol.31, No.4, pages 467–472, 1985. 396

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale election", *Advances in Cryptology – Auscrypt'92*, LNCS Vol.718, pages 244–260, Springer-Verlag, 1992. 389, 390

[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. 389, 390, 391, 392, 394, 403, 404

[HS00] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption", *Advances in Cryptology - Eurocrypt2000*, LNCS vol.1807, pages 539–556, Springer-Verlag, 2000. 390, 391, 392

[Jak98] M. Jakobsson, "A practical mix", *Advances in Cryptology – Eurocrypt'98*, LNCS Vol.1403, pages 449–461, Springer-Verlag, 1998. 390

[JSI96] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications", *Advances in Cryptology – Eurocrypt'96*, LNCS Vol.1070, pages 143–154, Springer-Verlag, 1996. 396

[LK00] B. Lee, and K. Kim, "Receipt-free electronic voting through collaboration of voter and honest verifier", *Proceeding of JW-ISC2000*, pages 101–108, Jan. 25-26, 2000, Okinawa, Japan. 389, 390, 391, 392

[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, pages 683–693, 2001. 390, 391, 392

[MH96] M. Michels and P. Horster, "Some remarks on a receipt-free and universally verifiable mix-type voting scheme", *Advances in Cryptology – Asiacrypt'96*, LNCS Vol.1163, pages 125–132, Springer-Verlag, 1996. 389, 390

[NR94] V. Niemi and A. Rendall, "How to prevent buying of votes in computer elections", *Advances in Cryptology – Asiacrypt'94*, LNCS Vol.917, pages 141–148, Springer-Verlag, 1994. 389

[Oka97] T. Okamoto, "Receipt-free electronic voting schemes for large scale elections", *Proc. of Workshop on Security Protocols'97*, LNCS Vol.1361, pages 25–35, Springer-Verlag, 1997. 391

[OMAFO99] M. Ohkubo, F. Miura, M. Abe, A. Fujioka and T. Okamoto, "An Improvement on a practical secret voting scheme", *Information Security'99*, LNCS Vol.1729, pages 225–234, Springer-Verlag, 1999. 390

[Pai99] P. Paillier, "Public-key cryptosystems based on discrete logarithms residues", *Advances in Cryptology - Eurocrypt '99*, LNCS Vol. 1592, pages 223–238, Springer-Verlag, 1999. 392

[Pfi94] B. Pfitzmann, "Breaking an efficient anonymous channel", *Advances in Cryptology – Eurocrypt'94*, LNCS Vol.950, pages 332–340, Springer-Verlag, 1994. 390

[PIK93] C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme", *Advances in Cryptology – Eurocrypt'93*, LNCS Vol.765, pages 248–259, Springer-Verlag, 1994. 390

[Po00] D. Pointcheval, "Self-scrambling anonymizers", *Proceedings of Financial Cryptography 2000*, Y. Frankel, Pages 259–275, LNCS 1962, Springer-Verlag, 2001. 390

[SK94] K. Sako and J. Killian, "Secure voting using partial compatible homomorphisms", *Advances in Cryptology – Crypto'94*, LNCS Vol.839, pages 411–424, Springer-Verlag, 1994. 390

[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 Vol.921, pages 393–403, Springer-Verlag, 1995. 390, 391