

Receipt-Freeness for Groth's e-Voting Schemes*

WEI HAN, KE-FEI CHEN AND DONG ZHENG

Department of Computer Science and Engineering

Shanghai Jiaotong University

Shanghai, 200240 P.R. China

Electronic voting is an important cryptographic application. Groth presented some efficient non-interactive zero-knowledge (NIZK) arguments based on homomorphic integer commitments for voting. He investigated four types of e-voting schemes: limited vote, approval vote, divisible vote and Borda vote. Receipt-freeness means that a voter is unable to construct a receipt to convince others she has voted for a particular candidate. It is a security property to protect the election against vote buying and coercion. Groth's schemes do not satisfy receipt-freeness for a voter can exploit the randomness she chooses in encryptions or commitments to construct a receipt. In this paper a receipt-free variant of the limited vote election protocol is constructed. A third party called "randomizer" is employed to re-encrypt the votes and to mask the commitments made by the voters while preserving the validity of the votes. The construction is generic and can be easily modified to introduce receipt-freeness into other types of Groth's e-voting schemes.

Keywords: electronic voting, receipt-freeness, homomorphic threshold encryption, designated-verifier proof

1. INTRODUCTION

Electronic voting has been researched for many years. There are four basic approaches: voting based on secret sharing [1, 2], voting based on mix-net [3], voting based on blind signature [4] and voting based on homomorphic encryption [5-7]. E-voting schemes based on homomorphic encryption do not need the complicated construction of the anonymous communication channels so they attract more concerns.

A practical electronic voting scheme should satisfy the following requirements:

1. *Eligibility*: only eligible voters can participate in the election.
2. *Privacy*: the content of the individual ballot is kept private. Only the final result is made public.
3. *Verifiability*: The tally process can be verified. The voting schemes based on blind signature only guarantee that the voter can verify her own vote is counted. Universal verifiability is preferred. Universal verifiability means that anyone including an independent third party can verify the tally.
4. *Robustness*: The voting system can tolerate a certain number of faulty participants, even if some voters or voting authorities cheat.
5. *Fairness*: No partial results are made public prior to the end of the ballot casting phase, as it may affect the voting result in some way.

Received May 16, 2007; revised August 23 & December 11, 2007; accepted January 31, 2008.

Communicated by Chi-Jen Lu.

* This paper was partially supported by the National Natural Science Foundation of China (No.60473020).

Additional security properties may be required in some cases. For instance, the *receipt-freeness* is a security property to thwart vote buying and coercion. This concept was first introduced by Benaloh and Tuinstra [8]. In a receipt-free election, a voter is unable to construct a verifiable receipt of her vote, so she cannot convince others that she casts a vote for a particular candidate. Sako and Kilian [9] presented a receipt-free mix-type voting scheme. But the scheme is inefficient for a cut-and-choose zero-knowledge proof is used. Lee and Kim [10] proposed a receipt-free election scheme based on homomorphic encryption by introducing a trusted third party called “honest verifier” (HV). A voter encrypts her vote, then HV re-randomizes the encryption to construct the final vote. However, Hirt found the voting protocols in [8] and [10] did not satisfy receipt-freeness. In [11] he showed how to construct receipts for the two protocols, and pointed out two basic ideas on how to construct a receipt-free electronic voting scheme: ballots shuffle by multiple authorities and ballot re-encryption by a randomizer. In the voting protocol [12] based on ballots shuffle, multiple authorities act like a mix-net. In order to hide the permutation of votes for different candidates, authorities in turn shuffle the encryptions of each valid ballot for a voter. In the voting protocol based on ballot re-encryption, an honest third party called “randomizer” re-randomizes the encrypted ballot for the voter, and proves to her that the final ballot is correctly re-encrypted in a designated-verifier manner. The employment of a randomizer can considerably improve the efficiency so it seems to be a better choice.

The availability of the untappable channel between the voter and the authority or between the voter and the randomizer is the weakest assumption for the currently known receipt-free election schemes. Some schemes assume the existence of two-way untappable channels but in essence a one-way untappable channel is enough. A direction of a one-way untappable channel can be turned around effectively to construct two-way untappable channels. By utilizing a one-time padding sent in the channel’s untappable direction, subsequent messages transferred in the other direction can be unconditionally and deniably encrypted. Note that the untappable channel implies the adversary cannot watch voter’s behavior at the very moment of voting. Otherwise there is no way to achieve receipt-freeness. With the development of hardware techniques, the tamper-resistant device, such as smart cards, can play the role of the randomizer. The channel through which the voter interacts with the smart card can be regarded as an internal channel so it is hard to tap for eavesdroppers. Lee and Kim [13] proposed a receipt-free electronic voting scheme with a tamper-resistant randomizer, but there was vote information leakage in their proofs.

Groth [14] presented efficient non-interactive zero-knowledge (NIZK) arguments for voting schemes based on homomorphic encryption. He investigated four types of elections: limited vote, approval vote, divisible vote and Borda vote. His voting schemes are not receipt-free for a voter can exploit the randomness in encryptions or commitments to construct a receipt. In this paper we present the first receipt-free variant of his limited vote election scheme by introducing a randomizer to re-randomize the voter’s encryptions and commitments. We give the design of the designated-verifier commitment masking proof and the joint proof between the voter and the randomizer. Along the similar line our method can be modified to introduce receipt-freeness into other types of Groth’s e-voting schemes.

2. PRELIMINARIES

2.1 Homomorphic Threshold Encryption and Homomorphic Integer Commitment

A probabilistic public-key encryption function: $E: P \times R \rightarrow C$ is homomorphic if for all $x_1, x_2 \in P$, $r_1, r_2 \in R$ it holds that $E(x_1; r_1) \boxtimes E(x_2; r_2) = E(x_1 + x_2; r_1 \oplus r_2)$, where P is a group which is the plaintext space, R is a group which is the randomness space, and C is a group which is the ciphertext space. The group operations in P , in R and in C are denoted by $+$, \oplus , and \boxtimes respectively. When the adversary generates a ciphertext C , an opening M, R and $e \neq 0$, so that $C^e = E(M; R)$ in a homomorphic cryptosystem, if we can find μ, ρ so that $M = e\rho$, $R = e\mu$ and $C = E(\mu; \rho)$, we call the homomorphic cryptosystem satisfying the root extraction property. Two instances of the homomorphic encryption schemes are “additive ElGamal” [5] and Paillier encryption [15]. Both of them are semantically secure, have the root extraction property and have threshold variants [7, 16, 17].

Some homomorphic integer commitment schemes [18-20] have been proposed. In this paper we use the same multi-exponentiation variant as that in [14]. We choose a modulus n as a product of two safe primes and random generators g_1, \dots, g_k, h of QR_n . To commit to integers m_1, \dots, m_k using randomness $r \in Z$, we compute $c = \text{com}(m_1, \dots, m_k; r) = g_1^{m_1} \dots g_k^{m_k} h^r \bmod n$. To open the commitment we reveal m_1, \dots, m_k, r . When the adversary comes up with a commitment c , an opening d_1, \dots, d_k and $e \neq 0$, so that $c^e = \text{com}(d_1, \dots, d_k; r)$ in a homomorphic commitment scheme, if we can compute $\mu_1, \dots, \mu_k, \rho$ so that $c = \text{com}(\mu_1, \dots, \mu_k; \rho)$ and $d_i = e\mu_i$, $i = 1, \dots, k$, we call the homomorphic commitment scheme satisfying the root extraction property. The commitment we described above has the root extraction property.

2.2 Zero-Knowledge Proof

In e-voting schemes based on homomorphic encryption, when a voter submits an encrypted ballot she must prove her ballot is correctly formed. She should give a zero-knowledge proof to ensure the validity of her vote while preserving the secrecy of it. In such schemes a three-move honest verifier zero-knowledge proof, also called the Σ – protocol [21], is usually employed. Assume we have a binary relation R consisting of pairs (x, w) , where we think of x as a public instance of a problem and w as a witness, a solution to the instance. Assume also that we have a three-move proof of knowledge of R : this protocol gets a string x as a common input for the prover and the verifier, whereas the prover gets w as a private input such that $(x, w) \in R$. Conversations in the protocol are of the form (a, e, z) , where the prover sends a , the verifier chooses e at random, the prover sends z , and the verifier decides whether to accept the claim that $x \in L$ where L is the language specified by the relation R . Such a protocol is said to be a Σ – protocol if it satisfies the following criteria:

- The protocol is *complete*: if the prover gets w as a private input such that $(x, w) \in R$, the verifier always accepts.
- The protocol is *special honest verifier zero-knowledge*: from a challenge e , one can efficiently generate a conversation (a, e, z) , with probability distribution equal to that of conversation between the honest prover and verifier where e occurs as a challenge.

- The protocol is *special soundness*: from the common input x and any pair of accepting conversations (a, e, z) , (a, e', z') where $e \neq e'$, one can compute w efficiently such that $(x, w) \in R$.

Using Fiat-Shamir heuristics [22] Σ – protocols can be transformed into the non-interactive form by employing a cryptographic hash function $Hash$ and letting the challenge be created as $e = Hash(x \| a)$. The symbol “ $\|$ ” denotes concatenation. The non-interactive proof is secure in the random oracle model [23]. The Σ – proofs can be combined in the “AND” or “OR” fashion in an efficient manner. More details can be found in [11].

2.3 Designated-Verifier Re-encryption Proof

The notion of designated-verifier proof was first introduced in [24]. How to use the designated-verifier re-encryption proof in the receipt-free voting schemes has been thoroughly discussed [11-13]. Here we review it for the completeness of the paper. A designated-verifier proof is a proof that is only convincing for the designated verifier, but it is completely useless when it is transferred to any other entity. The key idea is to prove the knowledge of whether the witness in question, or the secret key of the designated verifier. When the witness is the verifier’s secret key, the prover does not know it so the proof is convincing for the verifier, whereas, the fact that the verifier knows her secret key can enable her to simulate a proof indistinguishable from a proof indeed made by the prover.

In essence, designated-verifier re-encryption Σ – proofs can be constructed from the OR-combination of two Σ – proofs. One of the Σ – proofs is the identification scheme for the verifier. We select the Schnorr’s identification protocol [25] in this paper for it has a concise form. The parameters are as follows: let P be a prime, q a prime divisor in $P - 1$, and g an element of order q in Z_p^* . We suppose the verifier has chosen s_V in Z_q at random as her private key and has published $Z_V = g^{s_V} \bmod P$ as her public key. The other is a Σ – proof for the re-encryption. For a homomorphic encryption function E and a ciphertext e , we call e^* is a re-encryption of e if $e^* = e \boxtimes E(0; \xi)$ holds for a random integer $\xi \in R$ where R is the randomness space. Indeed the Σ – proof for the re-encryption is the Σ – proof that $e^* \boxminus e$ contains the plaintext 0 and the randomness ξ . The notation \boxminus denotes the inverse operation of \boxtimes . The designated-verifier re-encryption proof is depicted in Fig. 1. The non-interactive proof can be made by using the Fiat-Shamir heuristic. If ElGamal encryption is used, totally 12 modular exponentiations are needed for the re-encryption and the proof.

The proof in Fig. 1 comprises two proof conversations: (e', c_1, β) and (t_2, c_2, s_2) . The conversation (t_2, c_2, s_2) is a simulated Schnorr’s identification Σ – protocol by the prover. It has exactly the same probability distribution as a real conversation. We will prove (e', c_1, β) is a real Σ – proof conversation: (i) The witness is ξ . It is easy to see we have completeness. (ii) Regarding special honest zero-knowledge: let e_Δ denote $e^* \boxminus e$. Given a random c_1 if we choose β at random, the conversation $(E(0; \beta) \boxminus (e_\Delta)^{c_1}, c_1, E(0; \beta))$ is an accepting conversation. (iii) Regarding special soundness: assume we have two accepting conversations with the same first move: $(e', c_1^{(1)}, \beta^{(1)})$ and $(e', c_1^{(2)}, \beta^{(2)})$ with $c_1^{(1)} \neq c_1^{(2)}$. From the two verifying equations $E(0; \beta^{(1)}) = (e_\Delta)^{c_1^{(1)}} \boxtimes e'$ and $E(0; \beta^{(2)}) = (e_\Delta)^{c_1^{(2)}} \boxtimes e'$ we have $(e_\Delta)^{c_1^{(2)} - c_1^{(1)}} = E(0; \beta^{(2)}) \boxminus E(0; \beta^{(1)})$. By the root extraction property

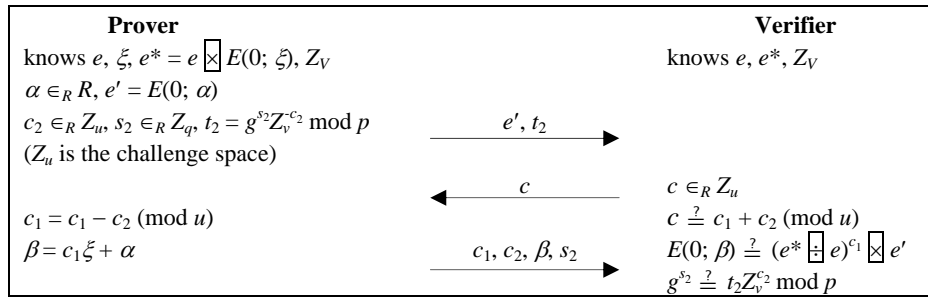


Fig. 1. Designated-verifier re-encryption proof.

of the homomorphic cryptosystem we can extract a witness μ so $E(0; \mu) = e_\Delta$. Now we can conclude that the proof in Fig. 1 is the OR combination of two Σ – proof conversations (e', c_1, β) and (t_2, c_2, s_2) .

Next we show the verifier can generate the re-encryption proof for any (e, e^*) using her knowledge of the secret key s_V such that $Z_V = g^{s_V} \bmod P$. The verifier selects $\tilde{\beta}, \tilde{c}_1$ and \tilde{r} at random, and computes $\tilde{e}' = E(0; \tilde{\beta}) \boxtimes (e^* \boxtimes e)^{\tilde{c}_1}$, $\tilde{c}_2 = c - \tilde{c}_1 \pmod u$, $\tilde{t}_2 = g^{\tilde{r}} \bmod P$, and $\tilde{s}_2 = c_2 s_V + \tilde{r} \bmod q$. The verifier sets $(\tilde{e}', \tilde{t}_2, c, \tilde{c}_1, \tilde{c}_2, \tilde{\beta}, \tilde{s}_2)$ as the proof. It is easy to see that this proof can pass the verification. The verifier can make the proof by herself so the designated-verifier re-encryption proof cannot be transferred to others.

2.4 Designated-Verifier Commitment Masking Proof

Consider the homomorphic integer commitment scheme $c = com(m_1, \dots, m_k; r) = g_1^{m_1} \dots g_k^{m_k} h^r \bmod n$, we call c^* is a commitment masking of c if $c^* = c \boxtimes com(0, \dots, 0; \xi)$ holds. Here in the commitment scheme we use the same operation notation \boxtimes as above just for notational convenience. The commitment masking is different from the re-encryption because the masked commitment c^* can no longer be opened by the original committer. c^* can only be jointly opened by the committer and the party who performs the masking operation. But the re-encryption can be decrypted by the holder of the private key alone without the help of the party who re-encrypts the ciphertext. This is the reason why we call it commitment masking instead of re-commitment. Luckily, it is unnecessary to open the commitments in Groth's zero-knowledge arguments for voting.

The Σ – proof for the commitment masking is the Σ – proof that $c^* \boxdiv c$ is a commitment to 0. For the commitment scheme $c = com(m_1, \dots, m_k; r) = g_1^{m_1} \dots g_k^{m_k} h^r \bmod n$, the prover indeed proves her knowledge of the secret exponentiation ξ of $h^\xi \bmod n$. The designated-verifier commitment masking proof is depicted in Fig. 2. Totally 8 modular exponentiations are needed for the commitment masking and the proof.

By using the similar techniques in the previous sub-section, it can be easily derived that the proof in Fig. 2 is the OR combination of two Σ – proof conversations (e', c_1, β) and (t_2, c_2, s_2) . Due to lack of space the complete proof is not given here. Note that the root extraction property of the commitment scheme is crucial to argue the special soundness. It can also be proved that the verifier is able to generate the commitment masking proof for any (c, c^*) using her knowledge of the secret key s_V .

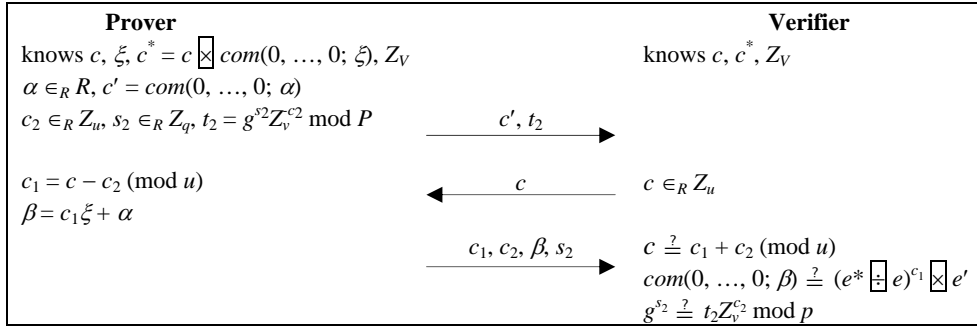


Fig. 2. Designated-verifier commitment masking proof.

3. A RECEIPT-FREE ELECTRONIC VOTING SCHEME

3.1 A Review on Groth's Election Schemes

In voting based on threshold homomorphic encryption, the voter must prove the ballot is correctly formed. In the early literature [5, 7, 11] the proofs followed the line of proving the encryption of the cast ballot is in the set of all valid ballots encryption. Such proofs can be made by the OR-combination of Σ – protocols. The OR combination is witness indistinguishable [26], so the prover can prove the validity of her vote while preserving the secrecy. But the overhead of this kind of proofs is rather high. Lipmaa et al. [27] proposed the first practical zero-knowledge argument based on homomorphic integer commitments. Using integer commitments they took advantage of special integer properties such as unique prime factorization. Damgard *et al.* [28] improved the scheme and presented a zero-knowledge argument for the limited vote election. Groth [14] generalized the NIZK arguments in [28] and constructed NIZK arguments for four types of electronic voting schemes: limited vote, approval vote, divisible vote and Borda vote.

We give a brief description on the Groth's NIZK argument for the limited vote election. The argument is shown in Fig. 3. The limited vote means that the voter can choose N distinct candidates from the fixed list including L candidates. We denote by M the strict upper bound on the number of votes that any candidate can receive. We select $M = p^2$ where p is a prime. The L candidates are represented with numbers $0, \dots, L - 1$ and a vote for N candidates i_1, \dots, i_N is encoded as $V = \sum_{j=1}^N M^{i_j}$. After a voter chooses the candidates, she encrypts her encoded vote and posts the encryption and the NIZK argument on the bulletin board. By the homomorphic property of the cryptosystem we can multiply all encrypted ballots $E(\sum_{j=1}^N M^{i_j})$ to get the encrypted form of the final result $E(\sum_{i=0}^{L-1} v_i M^i)$, where v_i is the number of votes on candidate i . The final result can be computed by threshold decryption.

We define the following parameters: We use a cryptographic hash function which has an l_e -bit output. The message space of the homomorphic encryption cryptosystem is Z_n for a suitable $n > M^L$, and we require n does not have prime factors smaller than 2^{l_e} . We assume l_v is the maximal bit-length of a vote. The randomizer spaces of the cryptosystem and the integer commitment are represented as an l_R -bit number and an l_r -bit

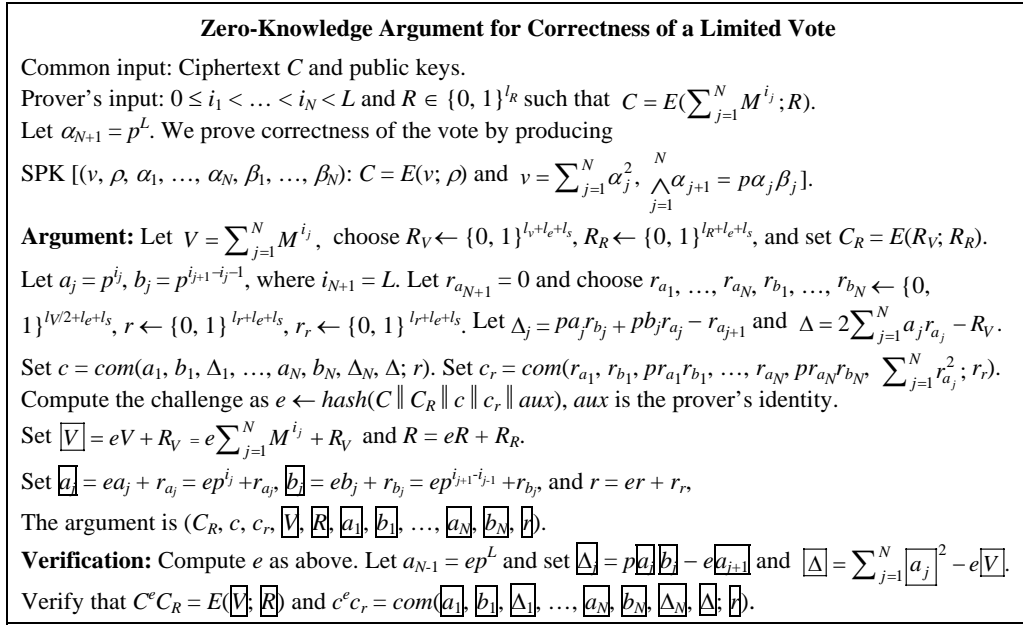


Fig. 3. Groth's NIZK argument for the limited vote.

number respectively. l_s is a security parameter such that for any value a we have $a + r_a$ and r_a are indistinguishable where r_a is a random $|a| + l_s$ -bit number.

In Groth's argument for the limited vote when a voter selects candidates i_1, \dots, i_N , she computes: $a_1 = p^{i_1}, a_2 = p^{i_2} = pa_1 b_1, \dots, a_N = p^{i_N} = pa_{N-1} b_{N-1}, a_{N+1} = p^L = pa_N b_N$. She encodes her vote as $V = \sum_{j=1}^N a_j^2$. If a limited vote is computed in the right form, the verification equation $c^e c_r = \text{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \dots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}; \boxed{r})$ will hold with overwhelming probability over the random choice on e . On the contrary, if the limited vote is encoded in the wrong form, the verification equation will hold with negligible probability. The verification equation $C^e C_R = E(\boxed{V}; \boxed{R})$ is used to check whether the voter has knowledge of the plaintext V in the encryption C . This check can prevent a dishonest voter from duplicating another voter's encrypted ballot as her vote. In essence, Groth's NIZK argument for the limited vote election scheme is the ADD-combination of Σ -proof for the internal arithmetic structure of the ballot and Σ -proof for plaintext knowledge.

3.2 Our Receipt-free Voting Scheme for the Limited Vote Election

We propose a receipt-free voting scheme for the limited vote election. The voting system consists of one registration authority and M talliers T_1, \dots, T_M . A threshold t denotes the least number of authorities that must remain honest and here we require $t > M/2$. Voters can choose N distinct candidates from a list including L candidates. A bulletin board is used to post data related to voting. The bulletin board is publicly accessible. Only the eligible participant can append messages in her own section but nothing can be deleted. There is certain authentication mechanism to control the access to the bulletin

board, such as digital signatures. Each voter can have access to a reliable third party called “randomizer”. The randomizer may be a server in the network, or the certified tamper resistant device such as a smart card. The smart card implementation is preferred since the computation can be distributed to each voter’s card and the card can be reused for many times. When the card is manufactured, the card owner’s identification information, such as the public key of the Schnorr’s identification scheme, should be fixed into the smart card. The randomizer has a signature key pair, of which the verification key is certified by the registration authority. We assume the existence of two-way untappable channels between the randomizer and the voter.

Stage 1: System Setup

M talliers T_1, \dots, T_M execute the key generation algorithm and publish the security parameters of the threshold (t, M) homomorphic cryptosystem and the homomorphic integer commitment scheme, such as public keys, on the bulletin board.

Stage 2: Registration

Each voter goes to the registration authority to register her identification information such as her name, public key, *etc.* The registration authority registers and certifies the signature verification key of the randomizer, and publishes the verification key on the bulletin board. If the randomizer is implemented with a smart card, the registration authority fixes the voter’s public key into the card, and issues the smart card to the voter. If the randomizer is implemented with a server in the network, the registration authority transfers the voters’ identification information to the randomizer through secure communication channels.

Stage 3: Voting

In this stage voter V_i and her randomizer jointly generate the encrypted ballot and the proof of the vote validity as follows:

1. V_i chooses N candidates from the list including L candidates. According to the Groth’s argument, she computes $V = \sum_{j=1}^N M^{i_j}$, $C = E(V, R)$, $C_R = E(R_V, R_R)$, $c = \text{com}(a_1, b_1, \Delta_1, \dots, a_N, b_N, \Delta_N, \Delta; r)$, $c_r = \text{com}(r_{a_1}, r_{b_1}, \text{pr}_{a_1} r_{b_1}, \dots, r_{a_N}, r_{b_N}, \text{pr}_{a_N} r_{b_N}, \sum_{j=1}^N r_{a_j}^2; r_r)$. She sends C, C_R, c, c_r to the randomizer with her signature through the untappable channels.
2. The randomizer verifies the signature, if correct, the randomizer re-encrypts C, C_R as $C^* = C \boxtimes E(0; R')$, $C_R^* = C_R \boxtimes E(0; R'_R)$. Then the randomizer masks the commitments c, c_r to get $c^* = c \boxtimes \text{com}(0, \dots, 0; r')$, $c_r^* = c_r \boxtimes \text{com}(0, \dots, 0; r'_r)$. The randomizer computes the designated-verifier re-encryption proofs and the designated-verifier commitment masking proofs for the voter V_i . The randomizer sends (C^*, C_R^*, c^*, c_r^*) and all the proofs to V_i through the untappable channels.
3. V_i verifies the proofs sent by the randomizer.
4. If (C^*, C_R^*, c^*, c_r^*) is generated correctly, V_i and the randomizer jointly compute the proof of the validity of the final vote. The process of the joint computation is depicted in Fig. 4. The fact that an honest randomizer will re-randomize the voter’s first messages randomly will make the output of the hash function into a random value. So in the computation of the hash function the prover’s identity does not need to be included.

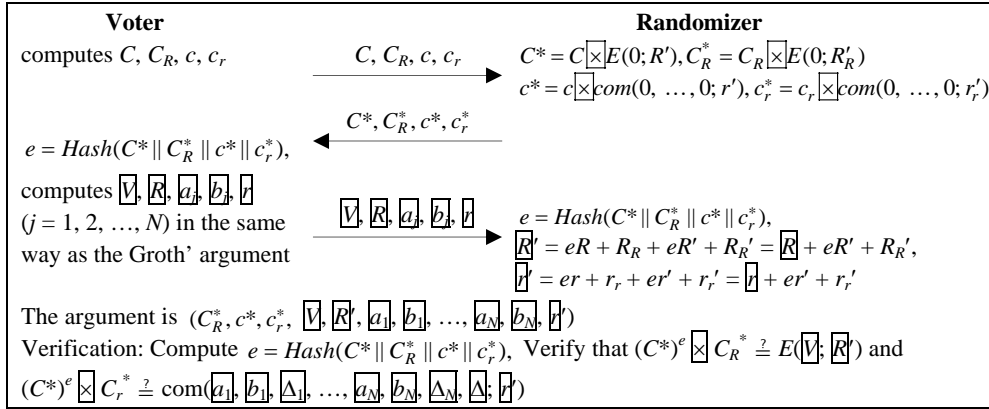


Fig. 4. The joint proof of the vote validity.

In Fig. 4 the randomizer plays the role of the verifier in the interaction between the voter and the randomizer, where the Σ – proof is the same as the interactive proof version of Groth's NIZK argument in Fig. 3. With knowledge of the witness the voter can correctly answer the challenge given by the randomizer if she constructs a valid encrypted ballot. In order to construct a non-interactive Σ – proof for the vote validity, the randomizer re-randomizes the initial message of the interactive Σ – proof conversation between the voter and the randomizer. Next, although the randomizer does not know the witness of the voter, it can use the voter as an oracle. Basing on the answer of the voter and its own randomness introduced into the initial message, the randomizer can compute a correct answer in the non-interactive Σ – proof. In the part of verification we have

$$(C^*)^e \boxtimes C_R^* = (C^e C_R)(E(0; R')^e E(0; R'_r)) = (C^e C_R)E(0; eR' + R'_r), \text{ and}$$

$$(c^*)^e \boxtimes c_r^* = (c^e c_r)(\text{com}(0, \dots, 0; r')^e \text{com}(0, \dots, 0; r'_r)) = (c^e c_r)\text{com}(0, \dots, 0; er' + r'_r).$$

If the voter is honest the equations: $C^e C_R = E(\boxed{V}; \boxed{R})$ and $c^e c_r = \text{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \dots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}, \boxed{r})$ will hold. We will have

$$(C^*)^e \boxtimes C_R^* = E(\boxed{V}; \boxed{R})E(0; eR' + R'_r) = E(\boxed{V}; \boxed{R}')$$

$$(c^*)^e \boxtimes c_r^* = \text{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \dots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}, \boxed{r})\text{com}(0, \dots, 0; er' + r'_r)$$

$$= \text{com}(\boxed{a_1}, \boxed{b_1}, \boxed{\Delta_1}, \dots, \boxed{a_N}, \boxed{b_N}, \boxed{\Delta_N}, \boxed{\Delta}, \boxed{r}').$$

In this proof, if the randomizer does not correctly perform the re-randomization or compute the final answer message, the voter can detect the failure by verifying the designated-verifier proofs or the joint proof. The voter can ask the registration authority to allocate another randomizer for her. The randomizer does not need to verify the third message of the voter because the voter's final vote will not be valid if the voter sends a bad third message. The voter can gain no information of the randomness R', R'_r, r', r'_r for she can only derive the value of $eR' + R'_r$ and $er' + r'_r$. In the non-interactive proof generated by the randomizer we use $\text{Hash}(C^* \| C_R^* \| c^* \| c_r^*)$ as the challenge because the zero-knowledge of proof is only honest verifier zero-knowledge. If we allow the randomizer

to ask any challenge, it may be possible for the randomizer to gain some information of the vote. The use of hash function ensures that the proof is zero-knowledge in the random oracle model.

The randomizer sends the final ballot C^* and the argument $(C_R^*, c^*, c_r^*, \mathbb{V}, \mathbb{R}', \underline{a}_1, \underline{b}_1, \dots, \underline{a}_N, \underline{b}_N, \mathbb{r}')$ and its signature to the voter.

5. The voter V_i verifies the argument and the signature. If they are correct, V_i posts C^* and the final argument $(C_R^*, c^*, c_r^*, \mathbb{V}, \mathbb{R}', \underline{a}_1, \underline{b}_1, \dots, \underline{a}_N, \underline{b}_N, \mathbb{r}')$ and the signature of the randomizer on the bulletin board.

Stage 4: Tallying

When the deadline of the election is reached, the registration authority collects all valid encrypted ballots, computes the product of them and posts the product on the bulletin board. A majority of talliers (such as t talliers or more) corporately decrypt the product, publish the final result and give a proof of the validity of the decryption on the bulletin board. Note that the secret key of the threshold homomorphic encryption cryptosystem is never reconstructed in the decryption process.

3.3 Security Analysis

The proposed electronic voting scheme satisfies the security requirements below:

Eligibility: The voter must register if she wants to participate in the election. Only legitimate voters can post messages on the bulletin board. The randomizer's signature verification key is registered and certified by the registration authority, so no other party can impersonate the randomizer.

Privacy: The vote is encrypted and the arguments are zero-knowledge in the random oracle model. If no less than t talliers remain honest, the single vote is never decrypted so the privacy of the individual vote is preserved. The randomizer can gain no information from the voter's proof.

Universal Verifiability: The final ballot and its related arguments are posted on the bulletin board. The validity of each ballot, the tally process and the joint decryption are publicly verifiable.

Robustness: The voter must prove her vote is in the right form, and (t, M) threshold encryption can tolerate the failure of maximum $M - t$ talliers.

Fairness: The partial result of the election will remain unknown because we have assumed that the number of honest talliers is no less than the threshold during the whole protocol execution.

Receipt-freeness: The designated-verifier re-encryption proof and the designated-verifier commitment masking proof made by the randomizer are sent to the voter through the untappable channels. These proofs cannot be transferred to others so the voter cannot

prove any relation between her first ballot and the final ballot. The voter cannot obtain the randomizer's internal randomness. In fact the voter has the ability to equivocate on her vote. We give a sketch for it:

Assume that $(C^{(1)}, C_R^{(1)}, c^{(1)}, c_r^{(1)})$ is the voter's first ballot, and (C^*, C_R^*, c^*, c_r^*) is the final ballot jointly constructed by the voter and the randomizer. The voter can "fake" another valid ballot $(C^{(2)}, C_R^{(2)}, c^{(2)}, c_r^{(2)})$ for other candidates, and she can easily make the following equations hold by selecting proper random numbers:

$$\begin{aligned} \boxed{V} &= eV^{(1)} + R_V^{(1)} = eV^{(2)} + R_V^{(2)}, \boxed{R} = eR^{(1)} + R_R^{(1)} = eR^{(2)} + R_R^{(2)} \\ \boxed{a_j} &= ea_j^{(1)} + r_{a_j}^{(1)} = ea_j^{(2)} + r_{a_j}^{(2)}, \boxed{b_j} = eb_j^{(1)} + r_{b_j}^{(1)} = eb_j^{(2)} + r_{b_j}^{(2)}, j = 1, 2, \dots, N, \\ \boxed{r} &= er^{(1)} + r_r^{(1)} = er^{(2)} + r_r^{(2)}. \end{aligned}$$

At that time the voter can claim that the randomizer uses $C^* \boxplus C^{(2)}, C_R^* \boxplus C_R^{(2)}$ to re-encrypt her ciphertexts and uses $c^* \boxplus c^{(2)}, c_r^* \boxplus c_r^{(2)}$ to mask her commitments. In fact $C^* \boxplus C^{(1)}, C_R^* \boxplus C_R^{(1)}, c^* \boxplus c^{(1)}$ and $c_r^* \boxplus c_r^{(1)}$ are used by the randomizer. The designated-verifier proofs can be simulated by the voter alone. Due to the semantic security of the homomorphic encryption, the hiding property of the commitment, the simulation property of the zero-knowledge proof and the untappability of the channels, the faked ballot $(C^{(2)}, C_R^{(2)}, c^{(2)}, c_r^{(2)})$ is indistinguishable from the ballot $(C^{(1)}, C_R^{(1)}, c^{(1)}, c_r^{(1)})$ to any third party including the vote-buyer or the coercer. So the scheme is receipt-free.

3.4 Efficiency Analysis

Compared with Groth's limited vote election scheme, the increased overhead of the voter results from the verification of the designated-verifier zero-knowledge proofs. The most increased overhead is distributed to the randomizer. The randomizer re-encrypts the ciphertexts C and C_R , masks the commitments c and c_r , and executes respective designated-verifier proofs. About 40 modular exponentiations in total are increased in the receipt-free voting scheme compared to the original limited vote election scheme if ElGamal encryption and the multi-exponentiation commitment are used. If the randomizer is implemented with tamper resistant device, the voter can interact with the randomizer without any network communication. This implementation is more convenient and practical.

4. CONCLUSION

Electronic voting will replace the ordinary paper based voting in the near future. The basic security properties and some additional properties, such as receipt-freeness, should be satisfied in order to put the e-voting into practice and gain people's confidence. Groth presented efficient NIZK arguments for voting and investigated four types of voting schemes. We propose a method to introduce receipt-freeness into his limited vote election scheme. A randomizer re-randomizes the voter's ballot and proves that the re-randomization is in the right form in a designated-verifier manner. The final ballot and

its proof are jointly generated by the voter and the randomizer. The similar ideas can be used to introduce receipt-freeness into other two types of Groth's voting schemes: approval voting and divisible voting. In Borda voting a shuffle argument is used. If the shuffle argument can be made divertible, our method is also applicable to Borda voting.

ACKNOWLEDGEMENT

The authors would like to thank Jens Groth and the anonymous referees for helpful suggestions.

REFERENCES

1. J. Benaloh, "Verifiable secret ballot elections," Ph.D. Thesis, Department of Computer Science, Yale University, New Haven, 1987.
2. B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Proceedings of Advances in Cryptology – CRYPTO*, LNCS 1666, 1999, pp. 148-164.
3. D. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms," *Communications of the ACM*, Vol. 24, 1981, pp. 84-88.
4. A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Proceedings of Advances in Cryptology – AUSCRYPT*, LNCS 718, 1992, pp. 244-251.
5. R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 1233, 1997, pp. 103-118.
6. O. Baudron, P. A. Fouque, D. Pointcheval, G. Poupard, and J. Stern, "Practical multi-candidate election system," in *Proceedings of the 20th ACM Symposium on Principles of Distributed Computing*, 2001, pp. 274-283.
7. I. Damgard and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," in *Proceedings of Public Key Cryptography*, LNCS 1992, 2001, pp. 119-136.
8. J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections," in *Proceedings of the 26th Annual Symposiums on the Theory of Computing*, 1994, pp. 544-553.
9. K. Sako and J. Kilian, "Receipt-free mix-type voting system – A practical solution to the implementation of a voting booth," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 921, 1995, pp. 393-403.
10. B. Lee and K. Kim, "Receipt-free electronic voting through collaboration of voter and honest verifier," in *Proceedings of Joint Workshop on Information Security and Cryptology*, 2000, pp. 101-108.
11. M. Hirt, "Mutliti-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, <ftp://ftp.inf.ethz.ch/pub/crypto/publications/Hirt01.pdf>.
12. M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 1807, 2000,

- pp. 539-556.
13. B. Lee and K. Kim, "Receipt-fee electronic voting scheme with a tamper-resistant randomizer," in *Proceedings of the 5th Information Conference on Information Security and Cryptology*, LNCS 2587, 2003, pp. 389-406.
 14. J. Groth, "Non-interactive zero-knowledge arguments for voting," in *Proceedings of International Conference on Applied Cryptography and Network Security*, LNCS 3531, 2005, pp. 467-482, <http://www.daimi.au.dk/~jg/ACNS05VoteProofFull.pdf>.
 15. P. Paillier, "Public-key cryptosystems based on composite degree residuosity class," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 1592, 1999, pp. 223-239.
 16. T. P. Pedersen, "A threshold cryptosystem without a trusted third party," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 547, 1991, pp. 522-526.
 17. P. A. Fouque, G. Poupard, and J. Stern, "Sharing decryption in the context of voting or lotteries," in *Proceedings of the 4th International Conference on Financial Cryptography*, LNCS 1962, 2001, pp. 90-104.
 18. E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," in *Proceedings of Advances in Cryptology – CRYPTO*, LNCS 1294, 1997, pp. 16-30.
 19. I. Damgard and E. Fujisaki, "A statistically-hiding integer commitment scheme based on groups with hidden order," in *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology – ASIACRYPT*, LNCS 2501, 2002, pp. 125-142.
 20. J. Groth, "Cryptography in subgroup of z_n^* ," in *Proceedings of Theory of Cryptography Conference*, LNCS 3378, 2005, pp. 50-65.
 21. R. Cramer, I. Damgard, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Proceedings of Advances in Cryptology – CRYPTO*, LNCS 839, 1994, pp. 174-187.
 22. A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proceedings of Advances in Cryptology – CRYPTO*, LNCS 263, 1986, pp. 186-194.
 23. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of ACM Conference on Computer and Communications Security*, 1993, pp. 62-73.
 24. M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated-verifier proofs and their applications," in *Proceedings of Advances in Cryptology – EUROCRYPT*, LNCS 1070, 1996, pp. 143-154.
 25. C. P. Schnorr, "Efficient signature generation for smart cards," *Journal of Cryptology*, Vol. 3, 1991, pp. 161-174.
 26. U. Feige and A. Shamir, "Witness indistinguishable and witness hiding protocols," in *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990, pp. 416-426.
 27. H. Lipmaa, N. Asokan, and V. Niemi, "Secure vickrey auctions without threshold trust," in *Proceedings of Financial Cryptography*, LNCS 2357, 2002, pp. 87-101.
 28. I. Damgard, J. Groth, and G. Salomonsen, "The theory and implementation of an electronic voting system," D. Gritzalis, ed., *Secure Electronic Voting*, Kluwer Academic Publishers, 2003, pp. 77-100.



Wei Han (韓璋) received his M.S. and B.S. degrees in the Department of Communication Engineering from Xidian University, Xi'an, in 2003 and 1999, respectively. He is a Ph.D. candidate in the Department of Computer Science of Shanghai Jiaotong University. His research interests include network security and e-commerce.



Ke-Fei Chen (陳克非) received his M.S. degree in Applied Mathematics from Xidian University, Xi'an, in 1985, and Ph.D. degree from Justus-Liebig University, Gießen, Germany, in 1994. He came to Shanghai Jiaotong University as associate professor in 1996, and assumed his present position as a professor in 1998. Dr. Chen has been concentrated his work in cryptography and information security. He is invited as a member of the Expert Panel Committee in the Department of Information Sciences of NSFC, he also is a member of the academic committee both of the State Key Laboratory of Information Security at the Chinese Academy of Sciences and the State Key Laboratory for Novel Software Technology at Nanjing University.



Dong Zheng (鄭東) received his Ph.D degree in Cryptography from Xidian University, Xi'an, in 1999. He is a Professor of the School of Information Security Engineering, Shanghai Jiaotong University. From 2002 to 2007, he was with the Department of Computer Science and Engineering, Shanghai Jiaotong University. His areas of research include provable security and new cryptographic technology, especially, coding theory and its application in secure distributed storage. His research is funded by NSFC, 863 and private companies.