PAPER *Special Section on Cryptography and Information Security*

A Secure and Practical Electronic Voting Scheme for Real World Environments

Wen-Shenq Juang[†], *Student Member and* **Chin-Laung Lei**[†], *Nonmember*

SUMMARY In this paper, we propose a practical and secure electronic voting scheme which meets the requirements of large scale general elections. This scheme involves voters, the administrator or so called the government and some scrutineers. In our scheme, a voter only has to communicate with the administrator three times and it ensures independence among voters without the need of any global computation. This scheme uses the threshold cryptosystem to guarantee the fairness among the candidate's campaign and to provide mechanism for achieving the function that any voter can make an open objection to the tally if his vote has not been published. This scheme preserves the privacy of a voter against the administrator, scrutineers, and other voters. Completeness, robustness, and verifiability of the voting process are ensured and hence no one can produce a false tally, corrupt or disrupt the election.

key words: privacy & security, secret ballot protocols, open objection, fair secret ballot schemes, uniquely blind signature.

1. Introduction

One of the hallmarks of democratic electoral systems is the institute of the secret ballot. Without ballot secrecy, the voters might be deterred from revealing their true opinions about the issues to be voted upon. In addition to the ballot secrecy, every interested voter must vote exactly once. Voting more than once can not be accepted by the administrator and other voters. Since electronic votes can be easily duplicated, there is a need to prevent malicious or careless voters from casting multiple votes. The naive approach of simply issuing a unique identification number to each voter would disclose privacy of the voters. To overcome this difficulty, many cryptographic protocols have been proposed [1]–[9]. Another feature in electronic voting scheme is that each voter can verify the voting result. When a voter finds that his vote has not been properly counted by the administrator, one approach is that via an anonymous channel he will broadcast his ballot to all voters. The validity of the result is based on the assumption that the anonymous votes are broadcasted correctly. The major drawback of this approach is its huge communication overhead.

In a typical real world election environment, there do not exist any single trusted party and the whole election process must be monitored by some chosen scrutineers. In this paper, we propose a secure and practical electronic voting scheme for real world voting environments with the following properties: (a) This protocol involves voters, the administrator or so called the government and some scrutineers; (b) If some voter finds his vote has not been properly counted, he can make open objection to the administrator via a public channel; (c) This protocol is fair, i.e., no one can get extra information about the tally result before the publication phase; (d) This protocol is collision free, i.e., a ballot of an eligible voter is always accepted by the administrator; (e) This protocol preserves the privacy of a voter against the administrator, scrutineers, and other voters; (f) It is robust in that no voter can disrupt or corrupt the election.

In this protocol, the computations among voters are independent without the need of any global computation and a voter only has to communicate with the administrator three times so this protocol is suitable for large scale general elections.

The remainder of the paper is organized as follows: In Section 2, previous works on secret ballot schemes are reviewed. In Section 3, we present our secret ballot protocol. The security considerations of this protocol are examined in Section 4. Then we discuss variants and possible extensions of our protocol in Section 5. Finally, a concluding remark is given in Section 6.

2. Related Work

Some boardroom voting schemes [4]–[6] have been proposed where voters openly send encrypted message back and forth until they all are confident of the outcome of the election. The major problems of these schemes are that the computations of voters are not independent and if any voter stops following the protocol during the voting, the election is disrupted. Nurmi *et al.* [3] proposed another secret ballot scheme based on ANDOS protocols [10]. For getting the administrator's secrets as ballots, voters need to communicate to each other. Fujioka *et al.* [1] proposed a secret ballot scheme which is more suitable for large scale elections since the computation and communication overheads are small even if the number of voters is large. To achieve the fairness property, every voter encrypts his vote by a random secret key and sends this encrypted vote to the counter through an anonymous channel [11], [12] in the voting phase. In the opening phase, to recover voters' intentions, every voter needs to send his random secret key

[†]The author is with the Dept. of Electrical Eng., National Taiwan University, Taipei, Taiwan.

to the counter through an anonymous channel. Any voter may not send his secret key to the counter and then the counter can not publish all voters' intensions. In this scheme, each voter must send two anonymous messages, but in the schemes [3], [7]–[9], each voter only has to send one anonymous message.

Iversen [2] proposed a voting scheme based on privacy homomorphism [13]. His scheme preserves the privacy of the voters against the administrator and other voters. The essential drawback of this scheme is that if all candidates conspire the privacy of the voters is violated. Moreover, this scheme is less practical for large scale elections since it requires a great deal of communication and computation when the number of voters is large. This protocol is not a general election protocol since the intentions of voters are only either "Yes" or "No".

The concept of blind signature was introduced by Chaum [14]. It allows the realization of secure voting schemes [1], [7]–[9] protecting the voters' privacy. Such systems have a party called the signer who is responsible for producing digital signatures. The other parties called requesters would like to obtain such signatures on messages they provide to the signer. A distinguishing property required by a typical blind signature scheme [14]–[16] is so-called the "unlinkability", which ensures that the requesters can prevent the signer from deriving the exact correspondence between the actual signing process performed by the signer and the signature which later made public. In a distributed environment, the signed blind messages can be thought as tickets in applications such as secret voting schemes. If the contents of the signed messages are the same, these signed messages will be thought as only one ticket. In [8], Juang *et al.* use the concept of blind signature and one-way permutations combined with voters' identifications to realize a uniquely blind signature scheme and proposed a collision free secret ballot protocol for computerized general election. This scheme is suitable for large scale general election. The essential drawback of this scheme is that if the administrator is not trustworthy, he may buy votes from the voters who have not registered before the publication of the result if necessary. Sako [7] proposed another approach in which voter can make his objection via a public channel, but her method can not hide the objector's privacy against the administrator.

The schemes proposed in [1], [3], [7] are not collision free, [3], [7], [8] are not fair, and [1]–[6] are not practical for large-scale elections.

Benaloh et al. [17] proposed a receipt free secret ballot protocol based on PEM (probabilistic encryption method) and the voting booths in which no more than a single voter can stay at the same time. In their protocol, anyone except the administrator can not coerce the voters into changing their intentions. Also, this protocol is not a general election protocol since the intention of any voter is only either "Yes" or "No". In this scheme, the privacy of any voter is preserved against others except the administrator.

3. The proposed voting scheme

In this section, a secure and practical electronic voting scheme for real world environments is presented. The protocol involves voters, the administrator and several scrutineers. The protocol consists of six phases: the initialization phase, the global key generating phase, the registration phase, the voting phase, the announcement phase and the publication phase. During the initialization phase, the administrator generates the system parameters. In the global key generating phase, all scrutineers cooperate to generate a threshold verifiable public key and distribute shares to each other without a trusted third party. In the registration phase, voters encrypt their intensions with the threshold public key generated in the global key generating phase and apply the uniquely blind signature technique to get their blind encrypted votes. In the voting phase, voters generate their real encrypted ballots from the blind encrypted votes received in the registration phase and send them to the administrator via an untraceable e-mail. In the announcement phase, the administrator publishes all accepted ballots. Finally, in the publication phase, if there does not exist any objection, the administrator first requests any k scrutineers for sending their shadow keys to him. When the administrator receives these k shadow keys, he computes the threshold secret key. Then he recovers voters' intentions and publishes all real ballots.

The underlying assumptions of this protocol are that: (a) Every eligible voter can communicate with the administrator and he can not abstain from the election process after the registration phase; (b) There exists a securely untraceable electronic e-mail system [11], [12]; (c) There exists a secure uniquely blind signature scheme and secure one-way permutation function [8]; (d) RSA cryptosystem is secure if factorization is intractable [18]; (e) ElGamal cryptosystem is secure if discrete logarithm problem is still intractable [19]; (f) At least $(m-k+1)$ scrutineers should not disclose their shadow keys before case 2 of the publication phase and k honest scrutineers must send their shadow keys to the administrator in the publication phase if no objection occurs.

In our protocol, every eligible voter can not abstain from the election process after the registration phase. Without this assumption, a malicious administrator can add extra votes as he wishes. All the single administrator election schemes [1], [3], [7], [8] have the above problem. In subsection 5.1, we will discuss how to solve this problem by several administrators.

Several anonymous channel protocols [11], [12], [20] have been proposed. The *mix-net* approach is used in

[11], [20] to realize a sender untraceable e-mail system. In the *mix-net* approach, the encrypted messages are sent to a *mix* agent who will disarrange all received messages and send them to the next agent. Finally, the last agent will send the encrypted messages to their destinations. The basic assumption of the *mix-net* approach is at least one mix agent is honest. In [21], Pfitzmann shows several attacks on the anonymous channels proposed in [20]. In the *mix-net* approach, it is harder to decide whether a voter has not sent his message to the administrator through an anonymous channel or the administrator does not receive it. In practical implementation, if there has audit records in the system, then this problem is solved. Otherwise, voters can send his message to the administrator and some trusted authorities via the *mix-nets*. The *dc-net* method based on the Dining Cryptographers Problem is used in [12] to achieve a sender untraceable e-mail system which is unconditionally or cryptographically secure depending on whether it is based on one-time keys or on keys generated by public key distribution systems or pseudo random number generators. Both mix-nets and dc-nets can be applied to our scheme, but we recommend mix-nets since we only need an email system which is periodic deliveries and not continuous deliveries.

Any secure uniquely blind signature scheme [8] is adequate for our scheme. The concept of blind signatures and one-way permutation functions combined with users' identifications are used in [8] to realize a uniquely blind signature scheme. For simplicity, we adopt RSA uniquely blind signature scheme in the following presentation. The existence of one-way permutations implies $P \neq NP$ [22]. Thus, no definitive one-way permutation has been found. The discrete logarithm function is a candidate that is believed by many researchers to be a one-way permutation [19], [23], [24].

In our scheme, we use the threshold cryptosystem [25] to preserve the fairness of the candidate's campaign and the function that any voter can make an open objection to the tally if his vote has not been published. In our proposed protocol, the public key authentication can be easily achieved by the $X.509$ directory authentication service [26]. The message authentication in our protocol is achieved by the RSA signature system in which the signed message m is attached with its signature $S(h(m \cdot RD))$, where S is the RSA signing function, h is a secure one-way hash function [26], [27] and RD is a redundancy string of the voting, against the multiplicative attack. The verification of the RSA signature can be achieved by the comparison method [28].

Assume that there are one administrator, n eligible voters and m scrutineers in this voting. Let e_{adm}, d_{adm}, n_{adm} be the RSA keys of the administrator, $e_{v,i}, d_{v,i}$ and $n_{v,i}$ be the RSA keys of eligible voter i , $e_{s,j}, d_{s,j}$ and $n_{s,j}$ be the RSA keys of scrutineer j. Let "$\cdot$" denote the ordinal string concatenation operator and $x \equiv_p y$ denote $x = y \bmod p$. Via the $X.509$ directory authentication service, any person can get every participant's public keys. For example, the RSA public keys of voter i are $(e_{v,i}, n_{v,i})$. Our proposed protocol is described in the following.

Phase 1 (the initialization phase)

The administrator randomly selects the RSA keys (e_a, d_a, n_a), where (e_a, n_a) are his public keys and d_a is his private key for this election. The administrator also selects the public threshold cryptosystem parameters (p, q, g) where p, q are two large prime numbers such that q divides $p-1$ and $g \equiv_p s^{(p-1)/q}$ $(gcd(s,p) = 1, s \neq 1)$, a public one-way permutation f, a public one-way function h, and the public redundancy bits RD for verifying the validity of each ballot. He also chooses a public constant k which is used in the k out of m threshold cryptosystem during the global key generating phase. Then he computes all signatures of these public parameters by his secret key d_{adm} and publishes these parameters and their corresponding signatures.

Phase 2 (the global key generating phase)

All scrutineers $j (1 \leq j \leq m)$, do the following:

1. Scrutineer j picks an ElGamal's key $(g^{a_j} \ mod \ p, -a_j)$, where $-a_j$ is the secret key and $g^{a_j} \ mod \ p$ is the public key, and chooses at random a polynomial $f_j(x) \equiv_q \sum_{i=0}^{k-1} f_{j,i} x^i$ of degree at most $k-1$ such that $f_{j,0} = a_j$. He then computes $GF_{j,i} \equiv_p g^{f_{j,i}}$ and the signatures $Cert\_GF_{j,i}$ on $GF_{j,i}$ for $0 \leq i \leq k-1$ and sends $(GF_{j,i}, Cert\_GF_{j,i})$ for $0 \leq i \leq k-1$ to the administrator.

2. Upon receiving all $(GF_{j,i}, Cert\_GF_{j,i})$ for $1 \leq j \leq m$ and $0 \leq i \leq k-1$, from all scrutineers, the administrator verifies if all $Cert\_GF_{j,i}$ are valid. If yes, he computes $G_p \equiv_p \prod_{i=1}^{m} GF_{i,0}$ and publishes $G_p, (GF_{j,l}, Cert\_GF_{j,l})$ for $1 \leq j \leq m$ and $0 \leq l \leq k-1$. Otherwise, he publishes the invalid signatures and then stops.

3. Scrutineer j sends $s_{j,l} \equiv_q f_j(l)$ and a signature $Cert\_s_{j,l}$ on $s_{j,l}$ secretly to every scrutineer $l (1 \leq l \leq m, l \neq j)$.

4. When scrutineer j receives all $s_{l,j} (1 \leq l \leq m, l \neq j)$ from other scrutineers, he verifies that the share $s_{l,j}$ received from scrutineer l is consistent with the published values $GF_{l,i}$ for $0 \leq i \leq k-1$ by verifying that $g^{s_{l,j}} \equiv_p \prod_{i=0}^{k-1} (GF_{l,i})^{j^i}$. If this fails, scrutineer j broadcasts that an error has been found, publishes $s_{l,j}$ and the signature $Cert\_s_{l,j}$ and the identification of scrutineer l and then stops. Otherwise, scrutineer j computes his share $s_j = \sum_{i=1}^{m} s_{i,j}$ and computes the signature $Cert\_GP_j$ on the threshold public key G_p. He then sends $Cert\_GP_j$ to the administrator.

5. Upon receiving all $Cert\_GP_j$ $(1 \leq j \leq m)$, the

administrator verifies if $Cert\_GF_j$ are valid for $1 \leqq j \leqq m$. If yes, he computes the signature $Cert\_GP$ on the threshold public key G_p and puts $(G_p, Cert\_GP, Cert\_GP_j(1 \leqq j \leqq m))$ on a public database. Otherwise, he publishes the invalid signatures and then stops.

Phase 3 (the registration phase)

Let ID_i be the identification of voter i. Voter i chooses two random strings R_i and φ_i. Voter i and the administrator then perform the following protocol.

1. Voter i computes the value $V_i = x_i \cdot \delta_i \cdot h(x_i \cdot \delta_i \cdot RD)$, where x_i is the intention of voter i and δ_i is a random number, chooses a random number r_i', computes the values $EV_i = (P_i \cdot Q_i) = (g^{r_i'} \cdot ((G_p^{r_i'})V_i \ mod\ p))$, $Reg_i \equiv_{n_{v,i}} (\varphi_i \cdot h(\varphi_i \cdot RD))^{d_{v,i}}$, $H_i = f(ID_i \cdot R_i)$, $M_i = H_i \cdot RD \cdot EV_i$, generates a random value $r_i (1 < r_i < n_a, gcd(r_i, n_a) = 1)$, computes $Y_i \equiv_{n_a} (r_i^{e_a} M_i)$, and finally sends $\Re_i = ((Y_i \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i}) \cdot ID_i$ to the administrator.

2. Upon receiving the message $\widehat{\Re_i}$, the administrator checks if $\widehat{\Re_i}$ is valid. If not, he will request voter i to retransmit the message $\Re_i$. When the administrator receives $\Re_i$, he checks the identification of voter i by verifying if $Reg_i^{e_{v,i}} \equiv_{n_{v,i}} \varphi_i \cdot h(\varphi_i \cdot RD)$. If not, the administrator rejects the registration of voter i. If yes and voter i has registered, the administrator also rejects the registration of voter i. Otherwise, he records the fact that voter i has registered by keeping $\Re_i$ in the registration database, computes $Z_i \equiv_{n_a} Y_i^{d_a}$, and sends Z_i to voter i.

Phase 4 (the voting phase)

Upon voter i receiving Z_i, he and the administrator do the following:

1. Voter i computes $X_i \equiv_{n_a} Z_i r_i^{-1} \equiv_{n_a} M_i^{d_a}$, and sends (X_i, M_i) anonymously to the administrator via an untraceable e-mail.

2. The administrator checks if $(X_i)^{e_a} \equiv_{n_a} H_i \cdot RD \cdot EV_i \equiv_{n_a} H_i \cdot RD \cdot (P_i \cdot Q_i) \equiv_{n_a} M_i$. If yes and RD is valid, he records (X_i, M_i). Otherwise he rejects it. He then sorts all (X_i, M_i) by M_i and preserves only one copy of M_i.

Phase 5 (the announcement phase)

The administrator publishes all the accepted ballot (X_i, M_i).

Phase 6 (the publication phase)

After the date time, there may occur two cases as follows:

1. Objection to published ballot

   Each voter has to check if his ballot has been published. If not, he broadcasts his encrypted ballot (X_i, M_i) to make an open objection.

2. Publishing the result

   If there is no objection, the administrator first requests any k honest scrutineers $Scru_{p_j}$ $(p_j \in [1, m], 1 \leqq j \leqq k)$ for sending their shadow keys s_{p_j} and computes the threshold secret key $G_s \equiv_q -\sum_{j=1}^{k} s_{p_j} \prod_{i=1, i \neq j}^{k} \frac{(-p_i)}{(p_j - p_i)}$. He recovers voter's intention from computing $V_i \equiv_p (P_i)^{G_s} Q_i$. He then publishes all ballots (X_i, M_i, V_i), all registrations $(\Re_j)$ and the threshold secret key G_s. Every person can check if every ballot is valid and the total number of the ballots is equal to the total number of the registrations to prevent that the administrator from adding any extra ballot to the tally.

4. Security

Keeping privacy of votes is the most important property of a secret ballot protocol. Also, the published tally must be equal to the actual result of the election, that is, each voter must vote exactly once and the administrator can not add extra ballots to the total tally. We now show that our proposed scheme possesses the above properties.

Definition 1 (Completeness)**:** A secret ballot protocol is said to be complete if the ballot of an eligible voter is always accepted by the administrator.

Before we show that our proposed scheme is complete, we first give the definition of a uniquely blind signature scheme.

Definition 2 (Collision freedom)**:** A uniquely (collision free) blind signature scheme is a blind signature scheme such that the signing function is injective and all the signatures requested by the honest requesters are distinct.

In our scheme, voter i sends a blind message $Y_i \equiv_{n_a} r_i^{e_a} M_i$, where $M_i = H_i \cdot RD \cdot EV_i$, $H_i = f(ID_i \cdot R_i)$, $1 < r_i < n_a$ and $\gcd(r_i, n_a) = 1$, to the administrator in step 1 of the registration phase. In step 1 of the voting phase, voter i can extract the blind signature $X_i \equiv_{n_a} M_i^{d_a}$. The role of the random string R_i is for increasing the security of the one-way permutation f. Since the entropy of user identifications ID_i is small, $H_i = f(ID_i \cdot R_i)$ is used to avoid the attack by an exhaustive search. It is clear that the signature scheme used in our proposed protocol is a uniquely blind signature scheme since this scheme is an RSA blind signature scheme [14] whose signing function is bijective and the signed message $M_i = H_i \cdot RD \cdot EV_i = f(ID_i \cdot R_i) \cdot RD \cdot EV_i$ is unique.

Based on the technique of uniquely blind signatures, we first show that our proposed scheme is complete.

Theorem 1: The secret ballot protocol of Section 3 is complete.

ProofThe proof is by contradiction. Assume that voter i follows the protocol and his vote is rejected by the administrator. In our protocol, the ballot (X_i, M_i) of voter i can only be rejected by the administrator either in step 2 of the registration phase or in step 2 of the voting phase.

(1) If the ballot of voter i is rejected in step 2 of the registration phase, there are two possibilities: (a) The administrator finds that $Reg_i^{e_{v,i}} \neq \varphi_i \cdot h(\varphi_i \cdot RD)$ and rejects his registration. Since every voter can communicate with the administrator, the administrator will receive $((Y_i \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i}) \cdot ID_i$ in step 2 of the registration phase. It clearly contradicts to the correctness of the RSA cryptosystem. (b) Assume that the administrator finds that voter i has registered in step 2 of the registration phase. Then there exists a malicious person that can forge $((Y_k \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i})$, where $Y_k \equiv_{n_a} (r_k^{e_a} M_k)$ and M_k, r_k are chosen by this malicious person, to impersonate voter i. It clearly contradicts to the assumption that the RSA signature scheme being secure.

(2) On the other hand, if the ballot of voter i is rejected in step 2 of the voting phase. Since voter i will follow the protocol, the administrator will receive his encrypted ballot (X_i, M_i) and record (X_i, M_i) in step 2 of the voting phase. Assume that there exists another voter j such that $H_j = H_i$ and then his ballot (X_i, M_i) is rejected by the administrator. Let ID_i be the identification of voter i, and f be the one-way permutation of the protocol. Since the identification of each voter is unique, we have $ID_i \cdot R_i \neq ID_j \cdot R_j$ for $i \neq j$. Thus, $H_i = f(ID_i \cdot R_i) \neq f(ID_j \cdot R_j) = H_j$. Contradiction. Therefore, we conclude that the secret ballot of Section 3 is complete. □

Definition 3 (Soundness)**:** A secret ballot protocol is said to be sound if no ineligible voters can vote.

In our protocol, an ineligible voter Alice can try to vote in the following possible ways.

In every election, the administrator chooses different RSA keys n_a, e_a and d_a. If the used ballots of previous election can be used again, Alice can forge the signatures made by the administrator. It clearly contradicts to the assumption that the RSA signature scheme being secure.

Second, if Alice can pass the check performed by the administrator in step 2 of the registration phase, he can forge $((Y_i \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i}) \cdot ID_i$. It clearly contradicts to the assumption that the RSA signature scheme being secure.

Third, if Alice can forge any valid ballot (X_k, M_k), where $X_k \equiv_{n_a} M_k^{d_a}$ and M_k is chosen by Alice, in step 1 of the voting phase, he can forge signatures generated by the administrator. It clearly contradicts to the assumption that the RSA signature scheme being secure.

From the above, our protocol is sound.

Next, we describe that no voter can vote more than once. In our scheme, only eligible voters can vote. In step 2 of the voting phase, the administrator will sort the ballots by M_i and preserve only one copy of all duplicate votes. If any eligible voter casts his ballot more than once, only one vote will be counted to the total tally. So no voter can vote successfully more than once.

In our protocol, any voter will vote exactly once. Also, it is desirable that the administrator can not add extra ballots to the total tally.

Definition 4 (Tally Correctness)**:** The result of a secret ballot protocol is said to be correct if the published tally is equal to the actual result of the election.

To show our protocol is correct, we will first establish a lemma which shows that any k honest scrutineers $Scru_{p_j}(p_j \in [1,m], 1 \leq j \leq k)$ can cooperate to reconstruct the threshold secret key G_s by their shadow keys $s_{p_j} (1 \leq j \leq k)$.

Lemma 1: Let $\psi(x) = \sum_{i=0}^{k-1} \psi_i x^i$ be the unique polynomial of degree at most $k-1$ such that $\psi(p_i) = s_{p_i} (p_i \in [i,m]$ and $1 \leq i \leq k)$. Then $G_s \equiv_q -\sum_{i=1}^{m} a_i \equiv_q -\psi(0)$.

ProofIn step 4 of the global key generating phase, after scrutineer j has received all $s_{i,j}$ $(1 \leq i \leq m, i \neq j)$, he verifies that the share $s_{i,j}$ received from scrutineer i is consistent with the published values $GF_{i,l}$ for $0 \leq l \leq k-1$ by verifying that $g^{s_{i,j}} \equiv_p \prod_{l=0}^{k-1} (GF_{i,l})^{j^l}$. So

$$g^{s_{i,j}} \equiv_p \prod_{l=0}^{k-1} \left(g^{f_{i,l}}\right)^{j^l} \equiv_p g^{\sum_{l=0}^{k-1} f_{i,l} * j^l}. \tag{1}$$

Since $g \equiv_p s^{(p-1)/q}$ and s is a generator of Z_p^*, g generates a cyclic subgroup S_g of Z_p^* with $|S_g| = q$. From (1), we can know that

$$s_{i,j} \equiv_q \sum_{l=0}^{k-1} f_{i,l} * j^l \tag{2}$$

Let $\mathcal{F}(x) = \sum_{j=1}^{m} f_j(x)$, where $f_j(x) = \sum_{i=0}^{k-1} f_{j,i}(x^i) \in Z_q(x)$ is the polynomial chosen by scrutineer j in step 1 of the global key generating phase. From (2) and step 4 of the global key generating phase, we can know that

$$s_j \equiv_q \sum_{i=1}^{m} s_{i,j} \equiv_q \sum_{i=1}^{m} \sum_{l=0}^{k-1} f_{i,l} * j^l \equiv_q \mathcal{F}(j). \tag{3}$$

From Lagrange polynomial theorem, given distinct k pairs (p_j, s_{p_j}) for $p_i \in [i,m]$ and $1 \leq i \leq k$, there exists a unique polynomial $\psi(x) = \sum_{i=0}^{k-1} \psi_i x^i$, such that $\psi(p_i) = s_{p_i} (p_i \in [1,m], 1 \leq i \leq k)$. So we can conclude that $\psi(x) = \mathcal{F}(x)$. And then it implies that $G_s \equiv_q -\sum_{i=1}^{m} a_i \equiv_q -\psi(0)$. □

Since our protocol is both sound and complete and no voter can vote successfully more than once, a voter

will vote exactly once. From *Lemma 1*, we know that if k out of m scrutineers are honest, the encrypted ballots will be opened correctly in the publication phase. The administrator must publish all registrations and ballots in the publication phase. In this protocol, every voter will follow the protocol and then the total number of the ballots must be equal to the total number of the registrations. Since every voter must check if his ballot has been counted properly and the total count of the registrations is equal to the total count of the published ballots, the administrator can not add any extra ballot to the tally. Therefore, the published tally is equal to the actual result of the election. It is clear that the result of secret ballot protocol of Section 3 is correct.

Definition 5 (Privacy)**:** A secret ballot protocol is said to be private if the privacy of voters is preserved.

In our protocol, a malicious person may try to derive the intention of voter i in the following possible ways: (1) Derive the link between the string $((Y_i \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i}) \cdot ID_i$ which is sent to the administrator in step 1 of the registration phase and the ballot (X_i, M_i, V_i) which is published in the publication phase. (2) Derive ID_i of voter i from his ballot (X_i, M_i, V_i) published in the publication phase. (3) Know where the source address of the ballot (X_i, M_i) sent to the administrator in step 1 of the voting phase is.

To derive the link between the string $((Y_i \cdot Reg_i)^{d_{v,i}} \bmod n_{v,i}) \cdot ID_i$ and the ballot (X_i, M_i, V_i) is computational infeasible since it clearly contradicts to assumption that the RSA uniquely blind signature scheme being secure.

To derive ID_i from the ballot (X_i, M_i, V_i) of voter i is computational infeasible since it clearly conflicts with the assumption that f is a one-way permutation function.

To derive where is the source address of the ballot (X_i, M_i) is computational infeasible since it clearly conflicts with the availability of a secure untraceable e-mail.

From the above, the secret ballot protocol of Section 3 is private.

Now, we want to show that the scheme satisfies the fairness property. Given the secret information of a group of l members ($0 \leqq l < k$), *Lemma 2* shows that the threshold cryptosystem constructed in the global key generating phase discloses no extra information about the threshold secret key G_s from the public information $\{GF_{i,j} | 1 \leqq i \leqq m, 0 \leqq j \leqq k-1\}$.

Lemma 2: Given a group of $l (0 \leqq l < k)$ members $G = \{p_i | p_i \in [1, m], 1 \leqq i \leqq l\}$ and the set of shares $\{s_{i,j} | 1 \leqq i \leqq m, j \in G\}$, for any fixed $i (1 \leqq i \leqq m)$ it can generate in polynomial time on $|q|$ a random set $\{g^{\widehat{f_{i,t}}} | 1 \leqq t \leqq k-1\}$ satisfying $g^{s_{i,j}} \equiv_p \prod_{t=0}^{k-1} (g^{\widehat{f_{i,t}}})^{j^t}$, for $j \in G$.

ProofFrom equation (2), we can know that given a fixed index i, the shares $s_{i,j} (j \in G)$ will use the same variables $\widehat{f_{i,t}} (0 \leqq t \leqq k-1)$ as follows:

$$s_{i,j} = \sum_{t=0}^{k-1} \widehat{f_{i,t}} * j^t. \tag{4}$$

Given a fixed index i, we can get at most l linear equations with $k (l < k)$ variables as follows:

$$s_{i,j} = \sum_{t=0}^{k-1} \widehat{f_{i,t}} * j^t (j \in G). \tag{5}$$

Since the linear equations have at least one solution $(\widehat{f_{i,t}} = f_{i,t}, 0 \leqq t \leqq k-1)$, we can solve the linear equations (5) and get a random solution $\widehat{f_{i,t}} (1 \leqq t \leqq k-1)$ by assigning random variables to all free variables. From (5), we can know that $g^{s_{i,j}} \equiv_p g^{\sum_{t=0}^{k-1} \widehat{f_{i,t}} * j^t} \equiv_p \prod_{t=0}^{k-1} (g^{\widehat{f_{i,t}}})^{j^t}$. □

Definition 6 (Fairness)**:** A secret ballot protocol is said to be fair if no one can get extra information of the tally result before the publication phase.

Theorem 2: The scheme proposed in Section 3 is fair.

ProofFrom *Lemma 2*, it is clear that the extra public information $\{GF_{i,j} | 1 \leqq i \leqq m, 0 \leqq j \leqq k-1\}$ is of no use to $l (0 \leqq l < k)$ scrutineers for deriving the threshold secret key G_s. The voting will not be affected since every registered voter i's encrypted ballot (X_i, M_i) must be published in the announcement phase and no votes can be added after beginning of the publication phase. By the assumption that ElGamal public key cryptosystem is secure, the proposed voting scheme is fair. □

The only way for a voter to disrupt the election is to make an open objection in the publication phase since every voter does not communicate to each other and only has to communicate with the administrator before the publication phase. In the voting phase, since either the data communication is recorded in the audit records or all the voters have to send their ballots to the administrator and a trusted authority, if some voter does not send his ballot to the administrator in the voting phase, he can not make objection to the administrator. Therefore, if the administrator is honest, no voter can disrupt the election. Furthermore, due to the fairness property of the proposed scheme, no one can get any partial information about the election before the publication phase.

5. Discussions

In the real world voting environments, there does not have a single trusted party and every candidate must be in fair campaigns, that is, no one can get any extra privilege from the voting process, so the voting process must be monitored by some scrutineers. In some critical situations, it is very hard to find any scrutineer. In these cases, every voter can play the role of scrutineer,

and join the global key generating phase. We now discuss how to make our scheme closer to the real world voting environment.

5.1 Distributing the power of a single administrator to several administrators

Since voters only need to communicate with the administrator in our protocol, there is no global computation among voters. But the administrator can impersonate the voter who abstains from voting after the registration phase. One of the basic assumptions of our protocol is that all the registered voters must cast their votes and no voter can abstain from voting. In real life, registered voters may abstain from voting after the registration phase. A simple approach to cope with this situation is that instead of sending the ballot to the administrator in the voting phase, voters send their ballots to a counter. Then some power of the administrator is distributed to the counter. If the administrator and the counter do not conspire and the probability of cases that voters abstain from voting is negligible, the administrator can not add extra ballots to the tally. Otherwise some modifications of the secret ballot system in Section 3 must be made. The modifications are described below: (1) Instead of a unique administrator, the modified system consists of κ administrators and at least one of them does not conspire with the others. (2) The voting protocol between each administrator and a voter is similar to the voting protocol in Section 3. (3) During the initialization phase, every administrator generates his RSA keys and all administrators agree the system common parameters. (4) In the publication phase, any interested voter must check if his vote has been properly counted. If his ballot is misplaced or not counted by any administrator, he broadcasts his encrypted ballot to make an open objection. If there is no objection, all administrators must request k honest scrutineers for getting the threshold secret key G_s, recover voters' intentions, and publish all real ballots. Anyone can check that the total numbers of the ballots published by all administrators are the same to prevent any malicious administrator from adding extra ballots to the tally.

By the above modifications, the power of a single administrator is distributed among several administrators and registered voters may abstain from voting after the registration phase. Also, Harn [29] proposed an efficient multisignature scheme based on the discrete logarithm problem. It is still an open problem that whether there exists an efficiently blind multisignature scheme. If this scheme exists, it can directly apply to our protocol to distributed the power of a single administrator to several administrators.

5.2 Make an open objection

In our scheme, there are two methods that an eligible voter can make objection to the tally as follows: (1) Through an untraceable e-mail, the voter broadcasts the encrypted ballot to all voters or sends it to a trusted party for making objection. (2) The voter broadcasts his open objection by sending his encrypted ballot.

Since there does not exist a single trusted party in some situations and the costs of communications via anonymous channel are higher than usual channels, we recommend that using the open objection method for making objection when administrator is cheating. In real world voting environments, the administrator's credit is very important. If voters find that the administrator has maliciously published a wrong tally result, then the voting process can be reinitialized.

6. Conclusion

In this paper, we propose a secure and practical election scheme for computerized general election which provides fairness, completeness, privacy, robustness, verifiability, and soundness properties. The most important property of this scheme is the fairness property. In our protocol, any voter can make an open objection to the tally without disclosing his privacy if his vote has not been published. In addition, our protocol is collision free. Our protocol is suitable for large scale general elections since the communication and computation overhead is small even if the number of voters is very huge.

References

[1] A. Fujioka, T. Okamoto, K. Ohta,"A practical secret voting scheme for large scale elections," Advances in Cryptology: Proc. of AusCrypt'92, LNCS 718, Springer-Verlag, pp. 244-251, 1992.

[2] K. R. Iversen, "A cryptographic scheme for computerized general elections," Advances in Cryptology: Proc. of Crypt'91, LNCS 576, Springer-Verlag, pp. 405-419,1991.

[3] H. Nurmi, A. Salomaa, L. Santean, "Secret ballot elections in computer networks," Computers & Security, Vol. 10, pp. 553-560, 1991.

[4] C. P. Pfleeger, "Security in computing," Prentice-Hall, Inc., 1989.

[5] A. Yao, "Protocols for secure communications," Proc. 23rd Annual IEEE Symp. on the Foundations of Computer Science, pp. 160-164, 1982.

[6] D. Chaum. "Elections with unconditionally secret ballots and disruption equivalent to breaking RSA," Advances in Cryptology: Proc. of EuroCrypt'88, LNCS 330, Springer-Verlag, pp. 177-182,1988.

[7] K. Sako, "Electronic voting scheme allowing open objection to the tally," IEICE Trans. fundamentals, Vol.E77-A, No.1, pp. 24-30. 1994.

[8] W. Juang, C. Lei, C. Fan. "A collision free secret ballot protocol for computerized general elections," to appear in Computers & Security (A preliminary version was presented at the 1994 Inter. Computer Symposium, Taiwan, pp. 309-

314.)

[9] W. Juang, C. Lei, C. Fan. "A secure and practical electronic voting scheme for real world environments," Proc. 6th National Conf. on Informa. Security, Taiwan, pp. 153-160, 1996.

[10] A. Salomaa, L. Santean. "Secret selling of secrets with many buyers," EATCS Bull., Vol. 42, pp. 178-186, 1990.

[11] D. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms," Commun. of the ACM, Vol. 24, No. 2, pp.84-88, 1981.

[12] D. Chaum. "The dining cryptographers problem: unconditional sender and recipient untraceability," J. of Cryptology, Vol. 1, pp. 65-75, 1988.

[13] K. R. Iversen. "A novel probabilistic additive privacy homomorphism," Proc. of the Inter. Conf. on Finite Fields, Coding Theory, and Advances in Communications and Computing, 1991.

[14] D. Chaum, "Blind signatures systems," Advances in Cryptology: Proc. of Crypt'83, Plenum, pp. 153.

[15] J. L. Camenisch, J. M. Pivereau and M. A. Stadler, " Blind signatures based on the discrete logarithm problem," Advances in Cryptology: Proc. of EuroCrypt'94, LNCS 950, pp. 428-432, Springer-Verlag, 1995.

[16] Chun-I Fan and Chin-Laung Lei, "Efficient blind signature scheme based on quadratic residues," Electronic Letters, Vol. 32, No. 9, pp. 811-813, 1996.

[17] J. Benaloh, D. Tuinstra, "Receipt free secret ballot elections," Proc. of the 26th Annual ACM Symp. on the theory of Computing, pp. 544-553, 1994.

[18] R. L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining digital Signatures and Public Key Cryptosystems," Commmun. ACM, pp. 120-126, 1978.

[19] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Inform. Theory, Vol. 31, pp. 469-472, 1985.

[20] C. Park, K. Itoh, K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme," Advances in Cryptology: Proc. of EuroCrypt'93, LNCS 765, Springer-Verlag, pp. 248-259, 1993.

[21] B. Pfitzmann, "Breaking an efficient anonymous channel," Advances in Cryptology: Proc. of EuroCrypt'94, LNCS 950, pp. 332-340, Springer-Verlag, 1995.

[22] M. R. Garey, D. S. Johnson,"Computer and intractability-a guide to the theory of NP-completeness," Murray Hill, 1979.

[23] W. Diffie, M. E. Hellman, "New directions in cryptography," IEEE trans. Inform. Theory. Vol. IT-22, pp. 644-654 ,1976.

[24] S. Pohlig, M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," IEEE Trans. on Inform. Theory, Vol. IT-24, pp. 106-110, 1978.

[25] T. P. Pedersen, "A threshold cryptosystem without a trusted party," Advances in Cryptology, Proc. of EuroCrypt'91, LNCS 547, Springer-Verlag, pp. 522-526, 1991.

[26] W. Stallings, "Network and internetwork security," Prentice hall international, pp. 267-282 & 333-340, 1995.

[27] Raiph C. Merkle, "One way hash functions and DES," Advances in Cryptology: Proc. of Crypt'89, LNCS 435, Springer-Verlag, 1990.

[28] T. Okamoto, "A Digital Multisignature Scheme Using Bijective Public Key Cryptosystem," ACM Trans. on Computer Sciences, Vol. 6, No. 8, pp. 432-441, 1988.

[29] L. Harn, "Group-oriented (t,n) threshold digital signature scheme and digital multisignature," IEE Proc. Comput. Digit. Tech., Vol. 141, No. 5, pp. 313, 1994.

Wen-Shenq Juang was born in Taichung, Taiwan in 1969. He received his BS degree in Computer Science and Information Engineering from Tatung Institute of Technology, Taiwan, in 1991, and M.S. degree in Computer Information Science from National Chiao Tung University, Taiwan, in 1993. He is now a Ph.D. candidate of electrical engineering at National Taiwan University. His current research interests include information security and cryptographic protocols in distributed environments. He is also a member of Chinese Cryptology and Information Security Association.

Chin-Laung Lei was born in Taipei, Taiwan in 1958. He received his BS degree in electrical engineering from National Taiwan University in 1980 and his Ph.D. degree in computer science from the University of Texas in 1986. From 1986 to 1988 he was an assistant professor of computer and information science at The Ohio State University. Since 1988, he has been an associate professor of electrical engineering at the National Taiwan University. His current research interests include cryptography and network security, parallel and distributed processing, operating system design, and design and analysis of algorithms. Dr. Lei is a member of the Institute of Electrical and Electronic Engineers, and the Association for Computing Machinery.