JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 865-877 (2001)

Practical and Efficient Electronic Voting Schemes*

ZHI-JIA TZENG AND WEN-GUEY TZENG
Department of Computer and Information Science
National Chiao Tung University
Hsinchu,300 Taiwan
E-mail: {zjtzeng, tzeng}@cis.nctu.edu.tw

We present various types of new electronic voting schemes, including two-way, multi-way and multi-selection election schemes, which guarantee privacy, universal verifiability, and robustness. Initially, a voter registers a polynomial function, his public key, with the election committee. Each voter uses his polynomial function to cast his vote in various elections. The *distinct feature* is that each term of a polynomial function corresponds to a candidate in a multi-way election. Thus, the final tally is independent among candidates, and the time complexity for searching final results is $O(n)$. In our schemes, each voter contacts the authorities only once; thus, our schemes are practical and suitable for large-scale elections. For robustness, we use the witness indistinguishable technique to construct the proof of validity. Security, then, is based on the discrete logarithm and decisional Diffie-Hellman assumptions.

Keywords: electronic vote, witness indistinguishable, multi-authority election, multi-selection vote, public verifiability, zero knowledge, secure multi-party computation

1. INTRODUCTION

Electronic voting protocols, which allow a set of voters to cast their votes for elections, are prime examples of secure multi-party computations. Through secure multi-party computations, voters seek to keep their inputs (votes) private. That is, other participants and outsiders learn nothing about the inputs of voters when they vote. Basically, a voting scheme should satisfy security requirements, such as privacy, universal verifiability, and robustness. Efficiency is an important criterion, too. In addition, we hope that a voting scheme will be practical and provably secure.

The voting schemes in [1, 3, 4, 8] are based on the *r*-th residuosity assumption. The scheme in [31] is based on the difficulty of the discrete logarithm. Cramer et al. [11] proposed efficient multi-authority elections for a voter. They used the threshold ElGamal cryptosystem in their schemes. The public key of the system is known to all voters, and the corresponding private key is shared with all authorities. Thus, each voter performs only one ElGamal encryption to cast his ballot. Furthermore, at least *k* honest authorities will exclude invalid ballots and tallies correctly to insure that ballots are valid. However, to authenticate the identity of a voter, they suppose that any public-key infrastructure, such as a digital signature, is already in place. These schemes achieve computational

Received December 26, 2000; accepted July 10, 2001.
Communicated by Chi Sung Laih.
* This work was supported in part by NSC grant 90-2213-E-009-145 and MOE Excellence grant 90-E-FA04-1-4, Taiwan, ROC.

security. In contrast to the above schemes, the scheme in [9] relies on information-theoretic security since it uses the property that many pairs (a_i, b_i) satisfy B = $g^{a_i} h^{b_i}$. However, this scheme is less effective since each voter needs to contact authorities for each election. Hevia and Kiwi [18] proposed electric jury voting schemes. Their schemes do not disclose the final tally of the votes but can determine whether the tally belongs to some pre-specified set. Recently, Katz et al. [23] proposed the notion of a cryptographic counter applied to electronic voting. Unlike the previous schemes, their counter does not rely on fully homomorphic encryption schemes. Their construction is based on the quadratic residuosity assumption. Many other reports [5, 6, 16, 27, 30] have also discussed voting schemes.

Receipt-freeness, introduced by Benaloh and Tuinstra [3], is another important property, A voter cannot carry away a receipt that proves how he voted. This may prevent vote buying or coercion. Currently, several incoercible schemes have been proposed [10, 25, 31]. Usually they assume that there is a physical constraint, such as a voting booth [3, 14, 25, 31]. The MIX net model was discussed in [15, 20, 21]. MIX net assumes that secret channels between voters and authorities exist. Recently, Hirt and Sako[19] proposed an efficient receipt-free voting scheme and showed that Benaloh and Tuinstra's schemes [3] are not receipt-free. Their scheme uses the techniques of homomorphic encryption and designated-verifier proofs [22]. In their scheme, each voter needs to contact authorities once for each election.

Our results. In this paper, we propose three electronic voting schemes: the multi-way, two-way and multi-selection schemes. Our proposals do not assume the existence of any public key infrastructure. Thus, a voter needs to contact authorities only once in the registration phase. Our schemes satisfy the essential requirements for voting schemes: privacy, robustness, and universal verifiability. The security of our schemes is based on the hardness of the decisional Diffie-Hellman and discrete logarithm problems. Our schemes do not have the property of receipt-freeness, which seems hard to achieve by means of public key based voting schemes. The features of our schemes are as follows.

- In our schemes, a voter contacts authorities to establish a secret polynomial once in the preparation phase and uses his secret polynomial to cast ballots for as many votes as the security parameter allows. For multi-way voting of t options, we use one term of the polynomial for an option. We can count the votes for each option in $O(n)$ time, where n is the number of valid ballots.
- Our schemes automatically authenticate voters such that they are secure against replay and impersonation attack. Therefore, one cannot copy the ballot of a voter.
- Since each voter goes through the preparation phase only once, our schemes are rather effective and suitable for large scale elections. In addition, we can extend our schemes to parallel and batch votes.
- Our schemes are efficient. For example, in multi-way voting of t options and m authorities, each voter takes k modular exponentiations to compute his public key in the preparation phase. For each ballot cast, a voter takes t modular exponentiations plus $t + 1$ non-interactive ballot proofs. When $t = 2$, our two-way voting scheme uses one modular exponentiation and one non-interactive ballot proof only for each ballot cast.

2. PRELIMINARIES

We describe the security requirements and the model for an electronic voting scheme in the following:

- **Privacy**. Each voter should be able to keep his vote secret unless more than k authorities together agree to open his vote. We say that an electronic voting scheme achieves computational privacy if some cryptographic assumption is required; otherwise, we say that the scheme achieves information-theoretic privacy.
- **Universal verifiability**. Each party, even an outsider, can verify the validity of a ballot. Therefore, invalid ballots can be discarded. Also, each party can check whether the final tally is consistent with the number of valid ballots.
- **Robustness**. An electronic voting scheme should be secure against a malicious coalition of a reasonable number of voters or authorities. Furthermore, it should be able to detect and discard illegal ballots; that is, no coalition of voters can disrupt the vote.
- **Eligibility**. Only eligible voters can cast valid ballots; that is, an unauthorized party cannot cast a valid ballot. Therefore, an electronic voting scheme should be able to defend against impersonation and replay attacks.
- **No vote duplication**. No voter can copy another voter's ballot; that is, a voter cannot cast a ballot without knowing the ballot's decision.
- **Receipt-freeness**. An electronic voting scheme should not allow a voter to carry a receipt that proves the way he voted. Otherwise, one could use the receipt for the purpose of vote buying and coercion.

Model. We assume that there is a *bulletin board* on which each involved party can publish some information. In our schemes, each eligible voter keeps his public key on the bulletin board. Each party's public key is not changed unless the corresponding private key is compromised. The bulletin board allows each party to publish and append messages to his designated section. No party can erase any information from the bulletin board. However, these published messages, which are related to some specific votes, will be erased by the voting committee after the vote is finalized.

There is a voting committee that controls all aspects of votes. We distribute the trust of the voting committee to m authorities, each of which possesses a share of a voter's secret. Our voting schemes have three phases: a preparation phase, ballot cast phase, and tally phase.

1. **Preparation phase**. For each election, the voting committee identifies the voters who possess the right to vote. Then, each new eligible voter sends his shares to the authorities and registers his public key on the bulletin board. Each old eligible voter uses his registered public key and shares, which are already on the bulletin board, and the authorities, respectively.
2. **Ballot cast phase**. For each election, the voting committee publishes a casting ballot parameter. Afterwards, each eligible voter uses the parameter to construct a ballot that consists of the vote and the validity proof of the vote. Finally, each voter posts his ballot on the bulletin board.

3. **Tally phase**. The authorities and each interested inspector check the validity of all the ballots. Invalid ballots shall be discarded. Then, the authorities publish some related parameters for tallying of the vote. Every party can tally the vote using the valid ballots and the published parameters.

3. THE BUILDING BLOCKS

We introduce some building blocks and definitions used in our schemes in this section. Let G_q be a group with prime order q. Typically, $G_q = \{i^2 \bmod p | i \in Z_p^*\}$, and $p = 2q + 1$.

3.1 Decisional Diffie-Hellman Assumption

The security of our schemes is based on the assumption that the decisional Diffie-Hellman (DDH) [2, 13] and discrete logarithm problems are hard. The DDH assumption states that no efficient (polynomial-time) algorithm can distinguish the following two distributions with a non-negligible advantage:

- the distribution **R** of random quadruples $\langle g, g^a, g^b, g^c \rangle \in G_q^4$;
- the distribution **D** of quadruples $\langle g, g^a, g^b, g^{ab} \rangle \in G_q^4$.

We can further prove that $\langle g, g^a, g^b, g^c \rangle$ and $\langle g, g^a, g^b, g^{(a+1)b} \rangle$ are polynomial-time indistinguishable, assuming that the DDH problem is hard.

3.2 Proof of Equality

Let g_1 and g_2 be generators in G_q, $h_1 = g_1^\alpha$, and $h_2 = g_2^\alpha$, where $\alpha \in Z_q^*$. The following protocol PROOF-EQ [12] proves that $\log_{g_1} h_1 = \log_{g_2} h_2$:

1. P sends $a_1 = g_1^w$ and $a_2 = g_2^w$ to V.
2. V randomly chooses $c \in_R Z_q$ and sends c to P.
3. P computes $r = w - \alpha \cdot c \bmod q$ and sends it to V.
4. V checks whether $a_1 = g_1^r \cdot h_1^c$ and $a_2 = g_2^r \cdot h_2^c$ hold.

Since PROOF-EQ is honest-verifier zero-knowledge, its non-interactive version NI-PROOF-EQ releases no useful information [11], where

NI-PROOF-EQ$(g_1, h_1, g_2, h_2) = (r, c)$

with $w \in_R Z_q$, $c = \mathcal{H}(g_1 \| h_1 \| g_2 \| h_2 \| a_1 \| a_2)$ and $r = w - \alpha \cdot c \bmod q$.

We can modify NI-PROOF-EQ to prove that the input has the form $\langle g_1, h_1, g_2, h_2 \rangle$ with $h_1 = g_1^\alpha$ and $h_2 = g_2^{\alpha+t}$. The protocol NI-PROOF-EQ-T is the same as NI-PROOF-EQ except that Step 4 is adjusted as

4′. V checks $a_1 = g_1^r \cdot h_1^c$ and $a_2 = g_2^r \cdot (h_2/g_2^t)^c$.

NI-PROOF-EQ-T is also special honest-verifier zero-knowledge.

3.3 Proof of a Ballot's Validity

In order to show that a voter's ballot is valid, the voter should construct a proof of the validity of his ballot. The proof is a witness indistinguishable protocol [17] such that every party can verify the ballot's validity but cannot compute the content of the ballot. Let h be a generator in G_q. The voter's ballot $v \in \{0, 1\}$ with respect to h is a tuple $\langle h, h^{a+v} \rangle$ plus a proof NI-PROOF-VOTE(g, g^a, h, h^{a+v}), where $a \in Z_q^*$. The protocol PROOF-VOTE(g, C, h, U) proves that

$$(\log_h U = \log_g C) \vee (\log_h U = \log_g C + 1).$$

PROOF-VOTE was proposed in [29]. We show it in the following for completeness.

1. P randomly selects w, r_{1-v}, $d_{1-v} \in_R Z_q$, computes $a_v = g^w$, $b_v = h^w$, $a_{1-v} = g^{r_{1-v}} C^{d_{1-v}}$ and $b_{1-v} = h^{r_{1-v}} (U/h^{1-v})^{d_{1-v}}$, and sends a_0, b_0, a_1, and b_1 to V.
2. V randomly selects c over Z_q and sends it to P.
3. P sets $d_v = c - d_{1-v} \bmod q$ and $r_v = w - ad_v \bmod q$, and sends d_0, r_0, d_1, and r_1 to V.
4. V verifies that $c = d_0 + d_1 \bmod q$, $a_0 = g^{r_0} C^{d_0}$, $b_0 = h^{r_0} U^{d_0}$, $a_1 = g^{r_1} C^{d_1}$, and $b_1 = h^{r_1} (U/h)^{d_1}$.

This protocol is also special honest-verifier zero-knowledge. We use NI-PROOF-VOTE(g, C, h, U) to denote its non-interactive version, where

$$\text{NI-PROOF-VOTE}(g, C, h, U) = (c, d_0, r_0, d_1, r_1)$$

with w, d_1, $r_1 \in_R Z_q$, $d_0 = c - d_1 \bmod q$, and $r_0 = w - ad_0 \bmod q$ if $v = 0$, and w, d_0, $r_0 \in_R Z_q$, $d_1 = c - d_0 \bmod q$, and $r_1 = w - ad_1 \bmod q$ if $v = 1$, and $c = \mathcal{H}(g \| C \| h \| U \| a_0 \| b_0 \| a_1 \| b_1)$ with $a_0 = g^w$, $b_0 = h^w$, $a_1 = g^{r_1} C^{d_1}$, $b_1 = h^{r_1} (U/h)^{d_1}$ if $v = 0$ and $a_0 = g^{r_0} C^{d_0}$, $b_0 = h^{r_0} U^{d_0}$, $a_1 = g^w$, $b_1 = h^w$ if $v = 1$. This protocol can be extended to the case where the vote can be 0 to $R - 1$, that is, to prove

$$(\log_h U = \log_g C) \vee (\log_h U = \log_g C + 1) \vee \ldots \vee (\log_h U = \log_g C + (R - 1)).$$

We use NI-PROOF-VOTE-R(g, C, h, U) to denote it.

4. THE VOTING SCHEMES

In this section, we will present three electronic voting schemes: the multi-way, two-way, and multi-election voting schemes.

4.1 Multi-Way Voting Scheme

We first present our multi-way voting scheme based on the building blocks discussed in Section 3. The system has m authorities such that k or more authorities together can reveal a ballot. In the preparation phase, the system sets up parameters, and each eligible voter registers to the system and selects a private key that is known to him only. Then, each eligible voter sends a share of his private key to each authority through a pri-

vate channel. The shares enable k or more authorities to tally the vote in the tally phase. In the ballot cast phase, each registered voter publishes his ballot on the bulletin board. After all eligible voters have cast their ballots or the due time has passed, the authorities together tally the vote publicly in the tally phase.

Note that the preparation phase is set up once for all votes. For each new vote, only the ballot cast and tally phases are executed.

- **Preparation phase**. Let m authorities be A_i, $1 \le i \le m$.
  1. Let G_q be a group of a large prime order q. The system selects and publishes a generator g of G_q.
  2. Each eligible voter u_i randomly selects a degree-(k-1) polynomial $f_i(x) = a_{i,0} + a_{i,1}x + \ldots + a_{i,k-1}x^{k-1} \bmod q$ (private key), registers his public key $(ID_i, g^{a_{i,0}}, g^{a_{i,1}}, \ldots, g^{a_{i,k-1}})$ to the system, and sends the share $(l, f_i(l))$ to authority A_l, $1 \le l \le m$. Each authority A_l verifies validity the of the share $(l, f_i(l))$ by means of $g^{f_i(l)} = \prod_{j=0}^{k-1} g^{a_{i,j}l^j}$.
- **Ballot cast phase**. Assume that there are t options to choose from for the vote, where $t \le k - 1$.
  1. The system publishes the vote parameter h that is a generator of G_q such that $\log_g h$ is unknown to all parties. Note that each vote uses a different h.
  2. Assume that voter u_i would like to choose option c, $0 \le c \le t - 1$. Then, the voter sets $v_{i,c} = 1$ and all other $v_{i,j} = 0$, $0 \le j \le t - 1$, $j \ne c$. Voter u_i posts
     (a) $(ID_i, h^{a_{i,0}+v_{i,0}}, \ldots, h^{a_{i,t-1}+v_{i,t-1}})$,
     (b) NI-PROOF-VOTE $(g, g^{a_{i,j}}, h, h^{a_{i,j}+v_{i,j}})$, $0 \le j \le t - 1$, and
     (c) NI-PROOF-VOTE $(g, g^{\sum_{j=0}^{t-1} a_{i,j}}, h, h^{\sum_{j=0}^{t-1} a_{i,j}+v_{i,j}})$
     on the bulletin board. This means that voter u_i can choose at most one of the t options.
- **Tally phase**. k honest authorities together count the vote in a distributed way. In fact, the set of authorities publishes some parameters so that each party can tally the vote. Without loss of generality, we assume that the honest authorities are A_1, A_2, …, A_k. The authorities do the following:
  1. They verify the proof of each voter's ballot and exclude the dishonest. Assume that there are n honest voters, $u_1, u_2, \ldots, u_n$.
  2. They compute h^{B_0}, h^{B_1}, ..., $h^{B_{t-1}}$ in a secure and distributed way, where $f_1(x) + f_2(x) + \cdots + f_n(x) = B_0 + B_1x + \cdots + B_{k-1}x^{k-1}$. This can be done by each authority A_l, who posts $h^{f_1(l)+f_2(l)+\ldots+f_n(l)}$, $1 \le l \le k$ on the bulletin board. One can compute h^{B_j} using the interpolation method, $0 \le j \le t - 1$.
  3. They produce a proof NI-PROOF-EQ$(g, \prod_{i=1}^{n} g^{a_{i,l}}, h, h^{B_l})$, $0 \le l \le t - 1$.
  4. The voting committee computes

     $$T_l = \prod_{i=1}^{n} h^{a_{i,l}+v_{i,l}} / h^{B_l} = h^{\sum_{i=1}^{n} v_{i,l}},$$

     $1 \le l \le t$. Since n is not huge, the committee can compare $h^{\sum_{i=1}^{n} v_{i,l}}$ with h^r for $0 \le r \le n$ and find $\sum_{i=1}^{n} v_{i,l}$, that is, the number of votes for option l.

Correctness. Through the following procedure, we can decide whether a voter is honest by verifying his NI-PROOF-VOTE proofs. For honest verifiers, the set of honest voters is the same, say $\{u_1, \ldots, u_n\}$. We can compute the product of their public keys by computing $\prod_{i=1}^{n} g^{a_{i,l}}$ for $0 \le l \le t-1$. Then, the authorities publish h^{B_j} for $1 \le l \le t-1$ and prove that the exponent B_j of h^{B_j} is equal to $\sum_{i=1}^{n} a_{i,l}$ for $0 \le j \le t-1$ by NI-PROOF-EQ. We can check if NI-PROOF-EQ is valid. Finally, we can correctly compute T_l for $1 \le l \le t$ and the final result $\sum_{i=1}^{n} v_{i,l}$ by exhaustive search.

Complexity. The analysis of complexity is as follows.

1. In the preparation phase, each voter needs k modular exponentiations to compute a public key. Each authority needs k modular exponentiations to verify the validity of a share. In particular, each voter uses private channel once to send shares to authorities.
2. In the ballot cast phase, each voter spends t modular exponentiations casting a ballot and $t + 1$ NI-PROOF-VOTE proofs for validity of a ballot for t options.
3. In the tally phase, the time spent verifying the proofs of each voter's ballot is proportional to the number of voters. The authorities compute t NI-PROOF-EQ proofs and spend $O(n)$ time searching the final results for n honest voters.

4.2 Two-Way Voting Scheme

Two-way voting is a special case of multi-way voting, but it can be simplified by using only the constant coefficient instead of two coefficients as in straightforward application of the multi-way voting scheme. Let 0 denote a "no" vote and 1 a "yes" vote. The preparation phase does not change.

- **Ballot cast phase**. The committee publishes the casting ballot parameter h, which is a generator in G_q and is independent of g. An eligible voter computes and posts his vote $(ID_i, h^{a_{i,0}+v_{i,0}})$, $v_{i,0} \in \{0, 1\}$ and NI-PROOF-VOTE $(g, g^{a_{i,0}}, h, h^{a_{i,0}+v_{i,0}})$, on the bulletin board.

- **Tally phase**. This is the same as in the multi-way voting scheme except that only h^{B_0} is needed.

4.3 Multi-Selection Voting Scheme

Our multi-way voting scheme can be extended to a multi-selection voting scheme in which a voter can cast more than one option. Assume that each voter can vote for at most R options among t options. Our scheme can be adjusted as follows.

- **Ballot cast phase**. This phase is almost the same as that in the multi-way voting scheme except that we use

  NI-PROOF-VOTE-R $(g, g^{\sum_{j=0}^{t-1} a_{i,j}}, h, h^{\sum_{j=0}^{t-1} a_{i,j}+v_{i,j}})$

  to replace

  NI-PROOF-VOTE $(g, g^{\sum_{j=0}^{t-1} a_{i,j}}, h, h^{\sum_{j=0}^{t-1} a_{i,j}+v_{i,j}})$.

• **Tally phase**. This phase is the same as that in the multi-way voting scheme.

Note that the tally result should be smaller than or equal to Rn.

5. SECURITY ANALYSIS

Our voting schemes satisfy three essential security requirements: privacy, universal verifiability, and robustness. In addition, our schemes are secure against impersonation even though every party is allowed to publish arbitrary messages on the bulletin board. Also, no vote duplication is possible in our schemes.

Suppose that all generators g and h are independent. To prove the privacy of our voting schemes, we reduce the DDH problem to the privacy problem of our schemes.

Lemma 5.1 (Privacy) Our schemes achieve computational privacy assuming that the DDH problem is hard.

Proof: It is easy to see that if $D' = (g, g^a, h, h^{a+1})$ and $R = (g, g^a, h, h^c)$ are indistinguishable, then the distributions R and D in the DDH problem are also distinguishable.

If an attacker A can distinguish a voter's decision with a non-negligible ϵ, he can compute v of (g, g^a, h, h^{a+v}) with a probability of $1/2 + \epsilon$. Therefore, A can distinguish D' and R, and thus, the DDH distributions with a non-negligible probability ϵ, which is a contradiction. ❑

Lemma 5.2 (Universal verifiability) Our voting schemes achieve universal verifiability assuming that the computing discrete logarithm is hard.

Proof: Each party can check the proof to verify a ballot's validity, check the proof of the "counting" public key to verify honest voters, and the correctness of the final tally, our schemes achieve universal verifiability. ❑

Lemma 5.3 (Robustness) Our voting schemes achieve robustness assuming that the discrete logarithm problem is hard.

Proof: A malicious voter cannot cast a bogus ballot since he has to produce a proof of the ballot's validity. If he can do so, he knows the secret of an eligible voter, which is a contradiction. Since less than k authorities cannot produce h^{B_l}, $0 \le l \le k-1$, the voting schemes can withstand a coalition attack of $k-1$ malicious authorities. Furthermore, each party can count and verify the tally result. Therefore, our schemes achieve robustness. ❑

Lemma 5.4 (Against impersonation) Our schemes are secure against impersonation assuming that the discrete logarithm problem is hard.

Proof: Assume that an adversary $\mathcal{A}$ can impersonate a voter U_i with public key $(g^{a_{i,0}}, g^{a_{i,1}}, ..., g^{a_{i,k-1}})$. Let the casting ballot parameter be h. The adversary $\mathcal{A}$ is

able to produce a proof of a ballot's validity with respect to h. Let us consider the proof NI-PROOF-VOTE $(g, g^{a_{i,j}}, h, h^{a_{i,j}+v_{i,j}})$. By the soundness property of NI-PROOF-VOTE, $\mathcal{A}$ should be able to pass different challenges c and c' for the same random (committed) value w; that is, A can produce two answers (d_0, r_0, d_1, r_1) and (d'_0, r'_0, d'_1, r'_1) for challenges c and c', respectively. We can compute the secret value $a_{i,j}$ from the two answers by $a_{i,j} = (r_0 - r_1)(d_1 - d_0)^{-1} \bmod q$, which is a contradiction of our assumption. ❑

Finally, our schemes prevent vote duplication because each eligible voter needs to register a public key with the voting committee. Since different ballot casting parameters are used for different votes, one cannot use the ballot of one vote for another vote. Therefore, only eligible voter can cast valid ballots for a particular vote.

6. DISCUSSION

Our voting schemes allow a voter to cast a ballot that says "no" to all options. We can force a voter to cast an "exact" vote for a multi-way vote. The method replaces

$$\text{NI-PROOF-EQ-1}\,(g, g^{\sum_{j=0}^{t-1} a_{i,j}}, h, h^{\sum_{j=0}^{t-1} a_{i,j}+v_{i,j}})$$

with

$$\text{NI-PROOF-VOTE}\,(g, g^{\sum_{j=0}^{t-1} a_{i,j}}, h, h^{\sum_{j=0}^{t-1} a_{i,j}+v_{i,j}}).$$

This guarantees that a ballot says "yes" for exactly one of the options.

Since h is different and independent for each vote, an outsider cannot simply replay previous messages as a valid ballot. It is important to select h independently. If we select an unsuitable h, i.e., a new h that is not independent of the old h, then the ballots will not achieve privacy, and a replay attack will be possible.

Our multi-way voting scheme can hold parallel votes. If all votes are two-way, we use each term $g^{a_{i,j}}$, $1 \le j \le k-1$, for each vote of a voter u_i and count them independently. If not all votes are two-way, then we can partition the k terms so that each partition corresponds to a vote. Note that we should use a proof for each vote. Assume that $g^{a_{i,j}}, g^{a_{i,j+1}}, ..., g^{a_{i,j'}}$ corresponds to a vote, we should provide the proof NI-PROOF-VOTE $(g, g^{\sum_{l=j}^{j'} a_{i,l}}, h, h^{\sum_{l=j}^{j'} a_{i,l}+v_{i,l}})$ for the vote. In this way, our scheme is very efficient.

Comparison. The multi-way election schemes [9, 11, 19], for k options, need $O(\sqrt{n}^{k-1})$ time to search all T_i to meet the final tally $W = G_1^{T_1} \cdots G_k^{T_k}$ using the baby-step giant-step algorithm for n voters, where G_i is a generator and T_i is the result of the election. When n is large, this is a significant factor. Furthermore, each G_i should be independent of G_j for all $i \ne j$; then, the final tally will be consistent. Otherwise, two or more sets of T_i will satisfy W. Therefore, Gs must be selected carefully. However, in our schemes, a voter only needs $O(n)$ time to search the final results. If g and h are not independent, then privacy will not be maintained.

7. A MODIFICATION

In this section, we will modify the multi-way voting scheme of Cramer et al. [11] such that the final tally becomes independent and more efficient since some time and storage ballots. We note that each voter needs a password to post messages on the bulletin board. The modified scheme is as follows.

- **Preparation phase**. Assume that there are l authorities. The system settings are as follows:
  1. The m authorities execute the distributed threshold ElGamal encryption scheme [11, 26] and publish $y = g^s$ as the system's public key. The secret s is shared by all the authorities such that A_i holds s_i. Each authority A_i publishes g^{s_i} as its public key on the bulletin board. At least k honest authorities are needed to decrypt a ciphertext.
  2. The vote casting parameter h is a generator in G_q.
- **Ballot cast phase**. Suppose that there are t options. Each voter u_j posts

  $(ID_i, g^{\alpha_j}, h^{v_j} y^{\alpha_j})$ and EL-PROOF $(g, g^{\alpha_j}, y, h^{v_j} y^{\alpha_j})$

  on the bulletin board, where for $1 \le j \le t$ and $v_j \in_R \{1, 0\}$, $v_j = 1$ means that voter u_i casts his vote for option j. EL-PROOF is similar to PROOF-VOTE and is shown in the Appendix. EL-PROOF $(g, \prod_{j=1}^{t} g^{\alpha_j}, y, \prod_{j=1}^{t} h^{v_j} y^{\alpha_j} / h)$ means that u_i says "yes" to at most one of the options. We can use NI-PROOF-EQ $(g, \prod_{j=1}^{t} g^{\alpha_j}, y, \prod_{j=1}^{t} h^{v_j} y^{\alpha_j} / h)$ to force u_i to vote for "exact" one of the options.
- **Tally phase**. The authorities check the proof of each ballot's validity and exclude invalid ballots. Only valid ballots are tallied. Suppose that there are n valid ballots of voters $u_1, u_2, \ldots, u_n$. The tally procedure is as follows.
  1. Let $(X_j, Y_j) = (\prod_{i=1}^{n} x_{i,j}, \prod_{i=1}^{n} y_{i,j}), 1 \le j \le t$, where $(x_{i,j}, y_{i,j})$ is voter u_i, who casts a partial vote for option j.
  2. Each authority A_i posts NI-PROOF-EQ $(g, g^{s_i}, X_j, X_j^{s_i})$ for $1 \le j \le t$.
  3. At least k honest authorities jointly decrypt (X_j, Y_j) by computing $H_j = Y_j / X_j^s$ for $1 \le j \le t$ using Lagrange interpolation.
  4. Let $H_j = h^{t_j}$ for $1 \le j \le t$ and compute t_j by means of exhaustive search.

Complexity. In the modified scheme, searching final results takes $O(n)$ time for n honest voters. The proof of validity and computation of (X_j, Y_j) can be done in parallel quite effectively. In particular, the result of each candidate is independent. Although each voter spends about t times more time than is used in the original scheme [11], the cost is still is quite low.

Theorem 7.1 If the decisional Diffie-Hellman and discrete logarithm problems are hard, the modified voting scheme achieves universal verifiability, computational privacy and robustness.

Proof: Since the proof is similar to that in [11], we omit it here. ❑

8. CONCLUSIONS

We have presented new electronic voting schemes using polynomial functions based on the hardness assumptions of the discrete logarithm and decisional Diffie-Hellman problems. The new schemes satisfy all security requirements except for receipt-freeness. Compared with other multi-authority voting schemes, our schemes are rather efficient. In particular, for two-way voting, each voter spends $O(1)$ cost casting a ballot. Furthermore, the final tally is exact, and searching final results takes $O(n)$ time. In the future, we will enable our schemes to achieve receipt-freeness.

APPENDIX

EL-PROOF(g, X, y, U) is to prove

$(\log_g X = \log_y U) \vee (\log_g X = \log_y (U/h))$,

which is described as follows:

1. P randomly selects w, r_{1-v}, $d_{1-v} \in_R Z_q$, computes $a_v = g^w$, $b_v = y^w$, $a_{1-v} = g^{r_{1-v}} X^{d_{1-v}}$ and $b_{1-v} = y^{r_{1-v}} (U / h^{1-v})^{d_{1-v}}$, and sends a_0, b_0, a_1 and b_1 to V.
2. V sends a challenge $c \in_R Z_q$ to P.
3. P computes $d_v = c - d_{1-v} \bmod q$ and $r_v = w - \alpha \cdot d_v \bmod q$, and sends d_0, d_1, r_0 and r_1 to V.
4. V checks whether $c = d_0 + d_1 \bmod q$, $a_0 = g^{r_0} X^{d_0}$, $b_0 = y^{r_0} U^{d_0}$, $a_1 = g^{r_1} X^{d_1}$ and $b_1 = y^{r_1} (U / h)^{d_1}$.

The protocol is witness indistinguishable if the DDH problem is hard. When $h = y$, EL-PROOF is equal to PROOF-VOTE.

REFERENCES

1. J. Benaloh, "Verifiable secret-ballot elections," PhD thesis, Department of Computer Science Department, Yale University, New Haven, 1987.
2. D. Boneh, "The decision Diffie-Hellman problem," in *Proceedings of the Third Algorithmic Number Theory Symposium*, Vol. 1423, 1998, pp. 48-63.
3. J. Benaloh, D. Tuinstra, "Receipt-free secret-ballot elections," in *Proceedings of the 26th Symposium on Theory of Computing, ACM*, 1994, pp. 544-553.
4. J. Benaloh and M. Yung, "Distributing the power of a government to enhance the privacy of voters," in *Proceedings of ACM 5th Symposium on Principles of Distributed Computing*, 1986, pp. 52-62.
5. D. Chaum, "Untraceable electronic mail, return address, anddigital pseudonyms," *Communications of the ACM*, Vol. 24, No. 2, 1981, pp. 84-88.
6. J. D. Cohen, "Improving privacy in cryptographic elections," manuscript, 1994.
7. R. Cramer, I. Damgard, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Proceedings of Advances in Cryptol-*

ogy-Crypto '94, 1994, pp. 174-187.
8. J. Cohen and M. Fischer, "A robust and verifiable cryptographically secure election scheme," in *Proceedings of the 26th Symposium on Symposium on Foundations of Computer Science*, *IEEE*, 1985, pp. 372-382.
9. R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung, "Multi-authority secret-ballot elections with linear work," in *Proceedings of Advances in Cryptology-Eurocrypt* '96, 1996, pp. 73-83.
10. R. Canetti and R. Gennaro, "Incoercible multiparty computation," in *Proceedings of the 37th Symposium on Symposium on Foundations of Computer Science*, *IEEE*, 1996, pp. 504-513.
11. R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Proceedings of Advances in Cryptology-Eurocrypt* '97, 1997, pp. 103-118.
12. D. Chaum and T. P. Pederson, "Wallet databases with observers," in *Proceedings of Advances in Cryptology-Crypto* '92, 1992, pp. 89-105.
13. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Proceedings of Advances in Cryptology-Crypto* '98, 1998, pp. 13-25.
14. I. Damgard and M. Jurik, "Efficient protocols based on probabilistic encryption using composite degree residue classes," manuscripts, 2000.
15. Y. Desmedt and K. Kurosawa, "How to break a practical MIX and design a new one," in *Proceedings of Advances in Cryptology-Eurocrypt* 2000, 2000, pp. 557-572.
16. A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Proceedings of Advances in Cryptology-Auscrypt* '92, 1992, pp. 244-251.
17. U. Feige and A. Shamir, "Wintness indistinguishable and witness hiding protocols," in *Proceedings of the 22th Symposium on Theory of Computing*, *ACM*, 1990, pp. 416-426.
18. A. Hevia and M. Kiwi, "Electric jury voting protocols," manuscripts, 2000.
19. M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *Proceedings of Advances in Cryptology-Eurocrypt* 2000, 2000, pp. 539-556.
20. M. Jakobsson, "A practical MIX," in *Proceedings of Advances in Cryptology-Eurocrypt* '98, 1998, pp. 448-461.
21. M. Jakobsson, "Flash mixing," in *Proceedings of ACM 18th Symposium on Principles of Distributed Computing*, 1998, pp. 448-461.
22. M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated-verifier proofs and their applications," in *Proceedings of Advances in Cryptology-Eurocrypt* '96, 1996, pp. 143-154.
23. J. Katz, S. Myers, and R. Ostrovsky, "Cryptographic counters and applications to electronic voting," in *Proceedings of Advances in Cryptology-Eurocrypt* 2001, 2001, pp. 78-92.
24. M. Michels and P. Horster, "Some remarks on a receipt-free and universally verifiable mix-type voting scheme," manuscript, 1996.
25. V. Niemi and A. Renvall, "How to prevent buying of votes in computer elections," in *Proceedings of Advances in Cryptology-Asiacrypt* '94, 1994, pp. 141-148.
26. T. P. Pedersen, "A threshold cryptosystem without a trusted party," in *Proceedings of*

Advances in Cryptology-Eurocrypt '91, 1991, pp. 522-526.
27. C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme," in *Proceedings of Advances in Cryptology-Eurocrypt* '93, 1993, pp. 248-259.
28. M. Stadler, "Publicly verifiable secret sharing," in *Proceedings of Advances in Cryptology-Eurocrypt* '96, 1996, pp. 190-199.
29. B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Proceedings of Advances in Cryptology-Crypto* '99, 1999, pp. 148-164.
30. K. Sako and J. Kilian, "Secure voting using partially compatible homomorphisms," in *Proceedings of Advances in Cryptology-Crypto* '94, 1994, pp. 411-424.
31. K. Sako and J. Kilian, "Receipt-free mix-type voting scheme–a practical solution to the implementation of a voting booth," in *Proceedings of Advances in Cryptology-Eurocrypt* '95, 1995, pp. 393-403.

Wen-Guey Tzeng (曾文貴) graduated from National Taiwan University in1985. He received his Master's and Ph.D. degrees in 1987 and 1991, respectively, from the State University of New York at Stony Brook. He joined the Department of Computer and Information Science, National Chiao Tung University, in 1991. Dr. Tzeng's current research interests include cryptology and information security.

Zhi-Jia Tzeng (曾志嘉) graduated from Chinese Culture University in 1995. He received his Master's degree in Computer and Information Science from National Chiao Tung University (NCTU) in 1997. He is currently a student at the Department of Computer and Information Science working on his Ph.D. thesis in the area of information security. His research interests include communication security, network security, and secure protocol design.