Secure E-Voting System with Collision-Free AID*

Lih-Chyau Wuu, Bae-Ling Chen, Lih-Woei Chen
Graduate School of Computer Science and Information Engineering
National YunLin University of Science & Technology
wuulc@yuntech.edu.tw

Abstract

This paper proposes an electronic voting system consisting of three roles: Voters, Authentication Center, and Tallying Centers; and proceeding to three phases: register phase, voting phase, and tallying phase. In the register phase, the voter cooperates with Authentication Center to generate a collision-free anonymous identity instead of voter's real identity to protect the voter's privacy and prevent from double-voting. We address a simple secret sharing scheme to split a ballot among the designated tallying centers that neither of them alone can restore the ballot. Such a scheme enforces the fairness property that no one can learn the voting outcome before the tally phase. In summary, we apply the collision-free anonymous identity technique and the secret sharing scheme to an e-voting system that the system is shown to have the secure properties: anonymity, eligibility, fairness, integrity, mobility, uniqueness and verifiability.

1. Introduction

Election is the main mechanism used to show the public opinion in a democratic society. The people in a country can express what they want via voting. An efficient and convenient voting environment usually increases the voting rate. However, the traditional paper-based voting always consumes lots of social cost. Therefore, over the past decades, a considerable numbers of researches have been made on secure *electronic voting* (e-voting). Chaum [5] used blind signature to make a requester obtain a signer's signature on a message without revealing the content of the message to the signer. Most e-voting systems [6, 8, 10, 13-15, 17, 20, 22, 24-27, 29-31, 34] require that ballots must be signed by some authorities before they are cast. Applying the blind signature technique to ballots can protect the contents of ballots from being revealed to the authority. Nevertheless, it is difficult to prevent from double voting if there is no other scheme to tie in with the blind signature in e-voting systems.

To prevent from double voting, a simple way is each cast ballot attached to a unique identity. However, to achieve the anonymity property that a ballot cannot be linked back to the voter who casts it, the voter must attach an anonymous identity instead of his/her real identity to the cast ballot. Most papers [6, 11, 13, 15, 17, 19, 22, 24, 34] claim that it is easy to have each voter with a unique anonymous identity but they do not state how to generate it. In this paper, we give a way to generate *collision-free anonymous identity* (AID), and apply it and a simple secret sharing scheme to an e-voting system with the following properties.

- **Anonymity:** A ballot can not be linked back to the voter who cast it.
- **Eligibility:** Only eligible voters can cast valid ballots.
- **Fairness:** No one can learn the voting outcome before the tally.
- **Integrity:** Each ballot is not able to be altered by anyone after it is cast.
- **Mobility:** Voters can vote anywhere.
- **Uniqueness:** No voter can vote more than once.
- **Verifiability:** Each voter can check if his/her ballot has been counted correctly.

The rest of this paper is organized as follows. Section 2 introduces the blind signature technique and the collision-free AID generating method which are used as the basic building blocks in the e-voting system. Section 3 describes how the e-voting system works. We analyze the proposed system and compare it with other related works [4, 6, 17, 21, 24] in Section 4. Finally, the conclusion is given in Section 5.

2. Basic Building Blocks

2.1 Blind Signature

Chaum [5] proposed the blind signature in 1981, which can prevent a signer from knowing the content of a signed message. Suppose that the signer B has an RSA key pairs of (PK_B, n) and (SK_B, n), where n is the product of two large primes. The following steps illustrate how a requester A obtains the blind signature of a message m from B without revealing the content of the message m.

Step 1. A sends a message $m' = m \cdot k^{PK_B} \pmod{n}$ to B, where $k \in_R Z_n^*$ is called blinding factor and $\mathbf{gcd}(k, n) \equiv 1$.

Step 2. After receiving the message m', B generates the signature $BS' = (m')^{SK_B} = (m \cdot k^{PK_B})^{SK_B} = m^{SK_B} \cdot k \pmod{n}$, and sends BS' back to A.

Step 3. A gets the desired signature by computing $BS = BS' \cdot k^{-1} = m^{SK_B} \pmod{n}$.

*This research was supported in part by the National Science Council of the R.O.C. under the grant NSC94-2213-E-224-028

2.2 Collision-Free Anonymous ID

The proposed method let a voter cooperate with the *Authentication Center* (AC) to generate a collision-free identity without revealing the content of the identity to the AC; such an ID is called *Anonymous Identity* (AID). It is noted that the AID also has the blind signature of the AC as the proof of a legal AID. Figure 1 illustrates the generating process. It is assumed that AC has an RSA key pairs of (PK_{AC}, n) and (SK_{AC}, n). p and q are two large primes, $q|(p-1)$, and g is a generator for Z_p^*. H is a secure one-way hash function with collision resistant property.

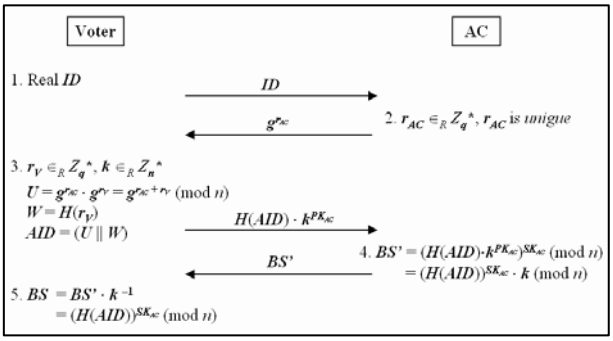

Fig. 1. Collision-free AID generating method

Step 1. AC authenticates an eligible voter.

Step 2. AC generates a unique $r_{AC} \in_R Z_q^*$ and sends the $g^{r_{AC}}$ to the voter.

Step 3. The voter randomly generates $r_V \in_R Z_q^*$ and computes $AID=(U\|W)$, where $U = g^{r_{AC}} \cdot g^{r_V} = g^{r_{AC}+r_V} \pmod n$, $W=H(r_V)$, and "||" denotes string concatenation operator. After that, the voter sends $H(AID) \cdot k^{PK_{AC}}$, where $k \in_R Z_n^*$, to require the AC to make a blind signature on it.

Step 4. AC signs on $H(AID) \cdot k^{PK_{AC}}$ and send the signature $BS' = (H(AID))^{SK_{AC}} \cdot k \pmod n$ back to the voter.

Step 5. The voter removes the blinding factor to get the AID signature, i.e., $(H(AID))^{SK_{AC}} \pmod n$.

Lemma 1. The proposed *AID* is collision-free.

Proof by Contradiction:

It is assumed that there exist two *AIDs*:

$AID = (U\|W) = g^{r_{AC}+r_V} \pmod p \,\|\, H(r_V)$,

$AID' = (U'\|W') = g^{r'_{AC}+r'_V} \pmod p \,\|\, H(r_V')$,

and $AID = AID'$. We consider two cases:

***Case* 1.** $r_V \neq r_V'$.

When $H(r_V) \neq H(r_V') \Rightarrow W \neq W' \Rightarrow AID \neq AID'$.

When $H(r_V) = H(r_V') \Rightarrow$ it is regarded as ***Case* 2**.

***Case* 2.** $r_V = r_V'$.

Since p, q are prime numbers, $q|(p-1)$, g is a generator for Z_p^*, r_{AC}, r_V, r_{AC}', $r_V' \in_R Z_q^*$ and *AC* generates different r_{AC} (i.e., $r_{AC} \neq r_{AC}'$) for each voter, so $g^{r_{AC}} \neq g^{r'_{AC}} \pmod p$. By **gcd**$(g^{r_V}, p)=1$, and $r_V=r_V'$, we get, $g^{r_{AC}+r_V} \neq g^{r'_{AC}+r_V} = g^{r'_{AC}+r'_V} \pmod p$, so $U \neq U' \Rightarrow AID \neq AID'$.

Thus the lemma holds. ❑

3. The Proposed E-Voting System

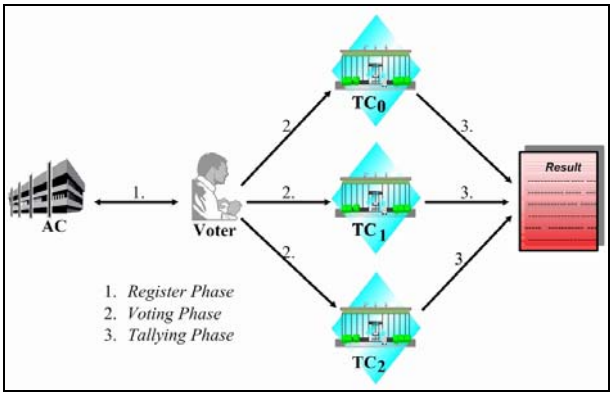

Fig. 2. The e-voting system structure

Figure 2 illustrates the structure of the e-voting system which consists of three participants: Voters, *Authentication Center* (AC), and *Tallying Centers* (TC_0, TC_1, TC_2). The system proceeds to three phases: *Register Phase*, *Voting Phase*, and *Tallying Phase*. Voter is the person who has the right to vote. Each voter must cooperate with AC to generate a collision-free AID to protect the voter's privacy and prevent from double voting. AC is responsible to cooperate with Voter to generate the voter's AID in the register phase, and make a blind signature for both of the voter's AID and ballot. The signature is not only as a proof that the voter's AID and ballot are legal, but also ensuring that no one can alter the cast ballot. To enforce the fairness property the ballot is spilt into three parts in the register phase and stored in each TC's database separately in the voting phase. All three TCs must cooperate to restore a complete ballot in the tallying phase. Table 1 illustrates the notations used in this paper.

Table 1. Notations

Notation	Description
V	Voter
AC	Authentication Center
TC	Tallying Center
AID	Anonymous ID
E	Election code
p, q	Two large primes (where $q\|(p-1)$)
g	A generator for Z_p^*
H	A secure one-way hash function with collision resistant property
(PK_X, n), (SK_X, n)	Long-term Public/Private key pair of user X
$Sig_X(Data)$	Data being accompanied with a digital signature given by user X; $Sig_X(Data) = Data \| (H(Data))^{SKX}$, where "|" denotes string concatenation operator
B	Ballot (with voter's intended candidate), *B is formed as a three-tuple of* (a_0, a_1, a_2), where $a_0 = x \oplus y \oplus B$, $a_1 = y \oplus z \oplus B$, $a_2 = x \oplus z \oplus B$
BS	AC's signature on voter's AID and ballot, $BS = (H(AID \| E) \| H(a_{i-1}) \| H(a_i^{PK_{TC_i}}) \| H(a_{i+1}))^{SK_{AC}} (\mathrm{mod} n)$, $i \in \{0, 1, 2\}$

We assumed that *AC* and three TCs are trustworthy and each participant has a certificate containing his/her real identity and his/her long-term public key. Before the register phase, *AC* announces the secure hash function *H*, the election code *E*, two large primes *p* and *q*, and the generator *g*, and AC has a database storing all certificates of eligible voters.

3.1 Register Phase

In the register phase, a voter must cooperate with AC to generate a unique AID. Although the AID is generated by the voter and AC together, only the voter knows the content of the AID. Our system uses the AID instead of the voter's real ID to protect the voter's privacy and prevent from double voting. The voter first transforms a ballot (*B*) into a three-tuple of (a_0, a_1, a_2) by XOR operation, and then encrypts one of (a_0, a_1, a_2) by the public key (PK_{TC_i}) of TC_i (i=0, 1, 2) to prevent an eavesdropper from restoring the complete ballot even though he/her collects all three parts of (a_{i-1}, $a_i^{PK_{TC_i}}$, a_{i+1}) of a ballot. After that, the voter requires AC to perform blind signature for the AID and the ballot before the ballot is cast. The signature is a proof that the cast ballot attached with the AID is legal. The following five steps **R1** ~ **R5** are atomic procedure and are executed in the register phase.

R1) $V \rightarrow AC$: $Sig_V(ID||E)$
A voter sends his/her real identity (*ID*) and the election code (*E*) to AC. Note that every election has a unique code to prevent from the replay attack. AC authenticates the voter by validating the signature attached on the message.

R2) $AC \rightarrow V$: $Sig_{AC}(g^{r_{AC}} \pmod p)$
After authenticating the voter, and checking the *ID* and *E*, the AC stores the *ID* in its *Register List* to prevent an eligible voter from double-register. Then AC generates a unique $r_{AC} \in_R Z_q^*$ and sends $g^{r_{AC}}$ back to the voter.

R3) $V \rightarrow AC$: $Sig_V(BH \cdot k^{PK_{AC}} \pmod n)$
The voter randomly selects a $r_V \in_R Z_q^*$ and computes $U = g^{r_{AC}} \cdot g^{r_V} = g^{r_{AC}+r_V} \pmod p$, $W = H(r_V)$, and $AID=(U||W)$. Then the voter makes a ballot (*B*) with his/her intended candidate and transforms *B* into a three-tuple of (a_0, a_1, a_2) by the following equations, where x, y, $z \in_R Z_n^*$ and $\oplus$ denotes XOR operator.

$a_0 = x \oplus y \oplus B$,
$a_1 = y \oplus z \oplus B$,
$a_2 = x \oplus z \oplus B$,

The voter randomly chooses $i \in \{0, 1, 2\}$ to encrypt a_i by using the public key (PK_{TC_i}) of TC_i to prevent the eavesdropper from restoring the complete ballot.
The voter computes

$$BH=(H(AID||E)||H(a_{i-1})|| H(a_i^{PK_{TC_i}}) ||H(a_{i+1}))$$

and blinds *BH* by computing $BH \cdot k^{PK_{AC}} \pmod n$, where $k \in_R Z_n^*$. After that, the voter sends $BH \cdot k^{PK_{AC}} \pmod n$ to AC for a blind signature. Note that the notation (i–1) or (i+1) throughout the paper imply '(i–1) **mod 3**' or '(i+1) **mod 3**' operation.

R4) $AC \rightarrow V$: $Sig_{AC}(BH^{SK_{AC}} \cdot k \pmod n)$
AC makes the blind signature for $BH \cdot k^{PK_{AC}} \pmod n$ by $(BH \cdot k^{PK_{AC}})^{SK_{AC}} = BH^{SK_{AC}} \cdot k \pmod n$ and sends $BH^{SK_{AC}} \cdot k \pmod n$ to the voter. Then AC stores (ID, r_{AC}, $BH^{SK_{AC}} \cdot k \pmod n$) in Register List.

R5) The voter gets AC's signature for his/her AID and ballot by computing

$$BS = BH^{SK_{AC}} \cdot k \cdot k^{-1} = BH^{SK_{AC}} \pmod n.$$

The signature *BS* is used to prove that the cast ballot attached with the AID is legal, as well as to ensure that no one can alter the content of the ballot because the digest of the ballot has been signed by AC.

3.2 Voting Phase

In the voting phase, the voter sends the following voting information depicted in step (**V1**) to each TC separately. Recall that in the register phase, a ballot (*B*) is transformed into a three-tuple of (a_0, a_1, a_2), and one of them is encrypted. That is, the cast ballot is formed as (a_{i-1}, $a_i^{PK_{TC_i}}$, a_{i+1}). The first byte of each message of step (**V1**) indicates that the corresponding a_i is encrypted or not, where '0' denotes 'no encryption' and '1' denotes 'encryption'. *BS* is the AC's signature on the voter's AID and ballet.

V1)
$$\begin{cases} V \rightarrow TC_{i-1} : (0, AID, BS, a_{i-1}, H(a_i^{PK_{TC_i}}), H(a_{i+1})) \\ V \rightarrow TC_i : (1, AID, BS, H(a_{i-1}), a_i^{PK_{TC_i}}, H(a_{i+1})) \\ V \rightarrow TC_{i+1} : (0, AID, BS, H(a_{i-1}), H(a_i^{PK_{TC_i}}), a_{i+1}) \end{cases}$$

Each TC prevents the voter from double voting by comparing the AID attached in the message of step (**V1**) with its *Ballot List*, and performs decryption to get $a_i = (a_i^{PK_{TC_i}})^{SK_{TC_i}}$ if the received message begins with '1'. Each TC validates the AID and the ballet by checking the following equation.

V2) $BS^{PK_{AC}} \stackrel{?}{\equiv} H(AID \,||\, E) \,||\, H(a_{i-1}) \,||\, H(a_i^{PK_{TC_i}}) \,||\, H(a_{i+1})$
If the equation holds, it means that the AID is legal and the partial ballet is correct. Then the TC_i stores the (AID, BS, a_i) into its Ballot List

3.3 Tallying Phase

In the tallying phase, all three TCs must cooperate to restore all ballots. A complete ballot (*B*) is restored by computing $B = a_0 \oplus a_1 \oplus a_2$. At the end of the tallying phase, TCs publish their Ballot Lists and the result of the election, and AC publishes its Register List that

each voter can check if his/her ballot has been counted correctly.

4. Analysis

In this section, we first analyze the security properties of the e-voting system, and then discuss the robustness of the system in defending against various attacks, and finally give a comparison of the system with the schemes proposed in [4, 6, 17, 21, 24].

4.1 Security Properties

- **Anonymity.** In the voting phase, a cast ballot attached with an AID instead of a voter's real ID. Although the AID is generated by the voter and AC together (AID= $g^{r_{AC}+r_V} \pmod p \,\|\, H(r_V)$), only the voter knows r_v. Thus, a ballot can not be linked back to the voter who cast it.
- **Eligibility.** It is assumed that AC has a database storing all certificates of eligible voters before the register phase. Our system requires that a voter must get AC's blind signature (*BS*) for his/her AID and ballot before voting. To do that, the voter makes a signature on the messages that AC authenticates the voter by validating the signature to make sure only the eligible voter can vote.
- **Fairness.** In the register phase, the ballot (*B*) is transformed into a three-tuple of (a_0, a_1, a_2), and one of them is encrypted by the public key (PK_{TC_i}) of TC_i. Even though the eavesdropper collects all three parts (a_{i-1}, $a_i^{PK_{TC_i}}$, a_{i+1}) of a ballot, he can not restore the ballot since he does not have the secret key of TC_i. This enforces the fairness property that no one can learn the voting outcome before the tally phase.
- **Integrity.** In register phase, only an eligible voter can get AC's signature on his ballot. The signature is $BS=(H(AID\|E)\|H(a_{i-1})\|\ H(a_i^{PK_{TC_i}})\ \|H(a_{i+1}))^{SK_{AC}}$. In Step (**V2**), each TC validates the cast ballot to make sure that no one can alter the ballot.
- **Mobility.** Since voters can vote anywhere via Internet, they are not restricted by the geographic locality.
- **Uniqueness.** AC maintains a register list to record which voters have performed the register phase that each voter can register once to get a collision-free AID only. In the voting phase, a ballot must be attached with a legal AID, and each TC maintains a Ballot List to record which AIDs have been used. Therefore no voter could vote more than once.
- **Verifiability.** The voter can confirm his/her ballot being correctly counted by checking the published Ballot List at tally phase. In addition, anyone can verify the outcome by checking $|Register\ List| \geqq |Ballot\ List|$.

4.2 Defending Against Various Attacks

- **The scenario of the replay attack.** The replay attack means that an eavesdropper collected messages during some election and replays such messages on the present election. The proposed system can prevent such replay attack. We assume that each participant uses the same long-term key pair for each election. E_{old} denotes the election code of some past election and E_{now} denotes the election code of the present election. We consider two cases:

  ***Case* 1.** In the register phase, if the eavesdropper replays the message of a voter ($Sig_V(ID\|E_{old})$, $Sig_V(BH \cdot k^{PK_{AC}})$), AC will find out $E_{now} \neq E_{old}$ in step (**R1**) and drop such a message since every election has a unique code (*E*).

  ***Case* 2.** In the voting phase, if the eavesdropper replay the message of a voter ((0, *AID*, *BS*, a_{i-1}, $H(a_i^{PK_{TC_i}})$, $H(a_{i+1})$), (1, *AID*, *BS*, $H(a_{i-1})$, $a_i^{PK_{TC_i}}$, $H(a_{i+1})$), (0, *AID*, *BS*, $H(a_{i-1})$, $H(a_i^{PK_{TC_i}})$, a_{i+1})), TC will find out $BS^{PK_{AC}} \neq H(AID\|E)\|H(a_{i-1})\|H(a_i^{PK_{TC_i}})\|H(a_{i+1})$ in step (**V2**) and drop such a message since every election has a unique code (*E*).

  Thus, the replay attack can not affect the e-voting system.

- **The scenario of a malicious voter.** If a malicious voter does not use the $g^{r_{AC}} \pmod p$ sent by AC in step (**R3**) to generate an illegal AID, and then gets the signature (*BS*) of the illegal AID from the AC in step (**R4**). It is possible that the illegal AID has the same value as a legal AID which is generated by a voter in a proper way. In the voting phase, we consider two cases:

  ***Case* 1.** The ballot attached with the illegal AID is cast before the one with the legal AID. In this case, the legal ballot will be rejected by TC, and the legal voter should send $Sig_V(ID\|r_v\|H(a_{i-1})\|\ H(a_i^{PK_{TC_i}})\ \|H(a_{i+1}))\|k)$ to AC. Since AC stored (*ID*, r_{AC}, $BH^{SK_{AC}} \cdot k \pmod n$) in Register List, AC first computes $AID=(g^{r_{AC}} \cdot g^{r_V})\| H(r_V)$ and then checks $BS^{SK_{AC}}.k \stackrel{?}{=} H(AID\|E)\|H(a_{i-1})\|H(a_i^{PK_{TC_i}})\|H(a_{i+1}).k$ If the equation holds, AC will inform TC to mark the ballot with the illegal AID useless, and AC will allow the legal voter to re-execute the register phase. As for the illegal voter, his/her ballot will not be counted in the tally phase since he/she does not following the proper way to generate the AID.

  ***Case* 2.** The ballot attached with the illegal AID is cast after the one with the legal AID. In this case, the illegal ballot will be rejected by TC, and the illegal voter can not have another AID since the

verification process described in ***Case* 1** can not succeed.

Thus the proposed system can prevent a malicious voter from generating his/her AID in a wrong way.

4.3 Comparison

In this section, we compare our e-voting system with the schemes proposed in [4], [6], [17], [21], and [24]. Table 2 illustrates that most of the schemes satisfy the security properties. The ballots of [6] could be altered if they were intercepted during the voting phase. The scheme of [17] does not satisfy the fairness property because the un-encrypting ballot is sent to TC in the voting phase; therefore TC can know the content of the ballot. The voting booths in [4] and [21] make those two systems not to have the mobility property. The system in [24] has all the E-voting properties as ours, however, the system needs additional devices.

Table 2. E-voting properties

	[4]	[6]	[17]	[21]	[24]	Ours
Anonymity	✓	✓	✓	✓	✓	✓
Eligibility	✓	✓	✓	✓	✓	✓
Integrity	✓	×	✓	✓	✓	✓
Fairness	✓	✓	×	✓	✓	✓
Mobility	×	✓	✓	×	✓	✓
Uniqueness	✓	✓	✓	✓	✓	✓
Verifiability	✓	✓	✓	✓	✓	✓

Table 3 dictates the computation overhead in each phase. The computation overhead includes the cost of exponentiation operation, multiplication/division operation, XOR operation and hash operation. The table shows our scheme needs only two XOR operations to recover each ballot in tally phase.

5. Conclusion

In this paper, a voter must cooperate with the AC to generate a collision-free AID, which is attached on the cast ballot to enforce the anonymity property and the uniqueness property. The integrity of a cast ballot is implemented by the blind signature of the AC, and the eligibility of a voter is checked by the AC. To enforce the fairness property, a simple secret sharing scheme is proposed to split a ballot among the three tallying centers that no one can learn the voting outcome before the tally phase. It is noted that only two XOR operations are needed to restore a ballot in the tallying phase. This indicates that the e-voting system can be implemented in real world and in large scale general election. Our future works will focus on distributing the overload of single AC into several ACs, and use (t, n) threshold scheme to split the ballot to make the e-voting system more robustness.

References

[1] M. Abe, "Universally Verifiable Mix-Net with Verification Work Independent of the Number of Mix-Servers," *Eurocrypt '98*, Helsinki, Finland, 1998, Page(s): 437-447.

[2] Baudron, Fouque et al. "Practical Multi-Candidate Election System," *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, Rhode Island, USA, 2001, Page(s): 274-283.

[3] J. Benaloh, and D. Tuinstra, "Receipt-free secret-ballot elections," *Proceedings of the 26th annual ACM symposium on Theory of computing*, Quebec, Canada, 1994, Page(s): 544-553.

[4] T. E. Carrol, and D. Grosu. "A Secure and Efficient Voter-Controlled Anonymous Election Scheme," *ITCC 2005*, Nevada, USA, 2005, Page(s):721 – 726

[5] D. Chaum, A. Sherman, and R. L. Rivest, "Blind signatures for untraceable payments," *Proc. of Advances in Cryptology-Crypto '82*, California, USA, 1982 Page(s): 199-203.

[6] Y. Y. Chen, J. K. Jan, and C. L. Chen, "The design of a secure anonymous Internet voting system," *ELSEVIER Computers & Security*, Vol. 23, 2004, Page(s): 330-337.

[7] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *Proceedings of Advances in Cryptology - Eurocrypt*, Konstanz, Germany, 1997, Page(s): 103-118.

[8] L. F. Cranor, and R. K. Cytron, "A security-conscious electronic polling system for the Internet," *30th Hawaii International Conference on System Sciences* (*HICSS*), 1997, Vol. 3, Page(s): 561-570.

[9] I. Damgard, and M. J. Jurik, "A generalisation, a simplification and some applications of paillier's probabilistic public-key system, "*proceedings of PKC '01*, 2001, Page(s): 119-136.

[10] G. Dini, "A secure and available electronic voting service for a large-scale distributed system," *ELSEVIER Future Generation Computer Systems*, 2003, Vol. 19, Page(s): 69-85.

[11] W. V. Gehrlein, and D. Lepelley, "The Condorcet efficiency of Borda Rule with anonymous voters," *ELSEVIER Mathematical Social Sciences*, 2001, Vol. 41, Page(s): 39-50.

[12] A. Hevia, and M. Kiwi, "Electronic jury voting protocols," *ELSEVIER Theoretical Computer Science*, 2004, Vol. 321, Page(s): 73-94.

[13] S. Y. Hwang, H. A. Wen, and T. Hwang, "On the security enhancement for anonymous secure e-voting over computer network" *ELSEVIER Computer Standards & Interfaces*, 2005, Vol. 27, Page(s): 163-168.

[14] S. Ibrahim, M. Kamat,M. Salleh, S.R.A. Aziz, "Secure E-voting with blind signature," *Telecommunication Technology, 2003* (*NCTT 2003*), Shah Alam, Malaysia, 2003, Page(s): 193 – 197.

[15] J. K. Jan, Y. Y. Chen, and Y. Lin, "The design of protocol for e-voting on the Internet," *Proceedings of IEEE 35th International Carnahan Conference on Security Technology*, London, UK, 2001, Page(s): 180 –189.

[16] J. K. Jan, and R. H. Lin, "A secure anonymous voting by employing DiffieeHellman PKD concept," *IEEE International Carnahan Conference on Security Technology*, Surrey, England; 1995. Page(s): 252-258.

[17] W. S. Juang, and C. L. Lei. "A collision-free secret ballot protocol for computerized general elections," *ELSEVIER Computer & Security*, Vol. 15, 1996, Page(s): 339-348.

[18] W. S. Juang, C. L. Lei and H. T. Liaw, "A verifiable multi-authority secret election allowing abstent from voting," *Computer Journal*, Vol. 45, 2002, Page(s): 672-682.
[19] J. Karro, and J. Wang, "Towards a practical, secure and very large scale online election," *Proceedings of 15th Annual Computer Security Applications Conference* (*ACSAC'99*), Arizona, USA, 1999, Page(s): 161-169.
[20] R. Kofler, R. Krimmer, A. Prosser, and M.K. Unger, "The role of digital signature cards in electronic voting," *Proceedings of the 37th Annual Hawaii International Conference on System Sciences* (*HICSS'04*), Hawaii, USA, 2004, Page(s): 1-7
[21] W. C. Ku, and C. M. Ho. "An e-Voting Scheme against Bribe and Coercion," *2004 IEEE International Conference on e-Technology, e-Commerce and e-Service* (*EEE'04*), Taipei, Taiwan, 2004, Page(s): 113 – 116.
[22] W. C. Ku, and S. D. Wang, "A secure and practical electronic voting scheme," *ELSEVIER Computer Communications*, Vol. 22, 1999, Page(s): 279–286.
[23] B. Lee, and K. Kim, "Receipt-Free Electronic Voting Scheme with a Tamper-Resistant Randomizer," *ICISC 2002*, Seoul, Korea, 2002, Page(s): 389-406.
[24] H. T. Liaw, "A secure electronic voting protocol for general elections," *ELSEVIER Computer & Security*, Vol. 23, 2004, Page(s): 107-119.
[25] I. C. Lin, M. S. Hwang, and C. C. Chang, "Security enhancement for anonymous secure e-voting over a network," *ELSEVIER Computer Standards & Interfaces*, 2003, Vol. 25, Page(s): 131-139.
[26] Y. Mu, and V. Varadharajan, "Anonymous secure e-voting over a network," *Proceedings of the 14th Annual Computer Security Applications Conference, CACSAC'98*, Arizona, USA, 1998, Page(s): 2936– 2939.
[27] T. Okamoto, "Receipt-Free Electronic Voting Schemes for Large Scale Elections," *Proceedings of Workshop on Security Protocols 1997*, Paris, France, 1997, Page(s): 25-35.
[28] T. P. Pedersen, "A threshold cryptosystem without a trusted party," *Proc. of Advances in Cryptology - Eurocrypt*, Brighton, UK, 1991, Page(s): 522-526.
[29] I. Ray, and N. Narasimhamurthi, "An anonymous electronic voting protocol for voting over the Internet," *Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems* (*WECWIS 2001*); California, USA, 2001, Page(s): 188-190.
[30] A. Riera, and J. Borrell, "Practical Approach to Anonymity in Large Scale Electronic Voting Schemes," *Proceedings of the 1999 Network and Distributed Systems Security Symposium* (*NDSS'99*), California, USA, 1999, Page(s): 69-82
[31] A. Riera, J. Rifa, and J. Borrell, "Efficient construction of vote-tags to allow open objection to the tally in electronic elections," *ELSEVIER Information Processing Letters*, 2000, Vol. 75, Page(s): 211-215.
[32] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," *Proceedings of Advances in Cryptology - Crypto '99*, California, USA, 1999 Page(s): 148-164.
[33] B. Schoenmakers, "Fully Auditable Electronic Secret-Ballot Elections," Xootic Magazine, Vol. 8, 2000, Page(s): 5-11
[34] S. H. Yun, and S. J. Lee, "An electronic voting scheme based on undeniable blind signature scheme," *Proceedings of the 37th IEEE Carnahan Conference on Security* (*ICCST*), Taipei, Taiwan, 2003, Page(s): 163–167.

Table 3. Computation overhead

	Register phase				**Voting phase**				**Tallying phase**			
Computations	**e**	**m**	**x**	**h**	**e**	**m**	**x**	**h**	**e**	**m**	**x**	**h**
[4]	×	×	×	×	6*l*+2	3*l*+1	0	0	*t*	*t*	0	0
[6]	2	2	0	1	3	0	1	2	2	1	1	0
[17]	4	1	0	1	1	1	0	0	×	×	×	×
[21]	14*L*	10*L*	0	0	4*L*-2	3*L*-1	0	1	*t*	*t*	0	0
[24]	×	×	×	×	6	3	0	1	2	1	0	0
Ours	5	3	6	5	4	0	0	6	0	0	2	0

e: exponentiation; **m:** multiplication/division; **x:** XOR; **h**: hash; ×: no computation overhead;
l: the number of selected participators; *t*: the number of tallying centers; *L*: the number of candidates.