ELSEVIER

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Standards & Interfaces 2190 (2002) 1–9

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Security enhancement for anonymous secure e-voting over a network

Iuon-Chang Lin[a], Min-Shiang Hwang[b,*], Chin-Chen Chang[a]

[a] *Department of Computer Science and Information Engineering, National Chung Cheng University, Chaiyi, Taiwan, ROC*
[b] *Department of Information Management, Chaoyang University of Technology, 168 Gifeng E. Road, Wufeng, Taichung County, 413, Taiwan, ROC*

Received 1 August 2002; received in revised form 19 November 2002; accepted 21 November 2002

Abstract

An electronic voting system makes it possible for the voters to cast their ballots over the computer network. Hence, voters can participate in elections without having to go to the polling places, which is more convenient and efficient. To design a practical voting scheme, Mu and Varadharajan have recently proposed an anonymous secure electronic voting scheme to be applied over the network. It does not only protect voters' privacy and prevent double voting, but also suits large-scale elections. However, the scheme has a weakness in security; that is, some voters may still double vote without being detected and may even reveal information they should not. In this paper, we shall show this weakness and improve the scheme to increase the protection against fraudulence.

Keywords: Blind signatures; Cryptography; Electronic voting; ElGamal public key cryptosystem

1. Introduction

Broadly speaking, electronic voting systems can be classified into two categories: without anonymous channel and with anonymous channel. In both voting systems, the voters can cast their ballots over networks such as the Internet. This is more convenient and efficient than conventional voting approaches because people can vote without showing up in the polling places. Furthermore, the systems cannot invade voters' privacy. The intention of each voter must be kept secret. To deal with the property of privacy, the first type uses the framework of multi-authority to hide the content of each vote [7,8]. Secret sharing schemes and zero-knowledge proof are often used to guarantee that no single authority can determine any individual vote. However, the voters have to communicate with several authorities in the voting process, and the system requires more computation in the vote counting procedure. This is unfavorable especially when it comes to large-scale elections. Therefore, to design a practical and efficient voting scheme for large-scale elections, the choice is usually with anonymous channel.

Having consulted several anonymous voting schemes [6,9,12], we decide on the privacy protection type as hiding the voters' identities and delivering the

* Corresponding author. Fax: +886-4-23742337.
E-mail address: mshwang@cyut.edu.tw (M.-S. Hwang).

0920-5489/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
doi:10.1016/S0920-5489(02)00120-4

contents of the votes to the authority publicly. Such systems must satisfy the following requirements [14,15].

- Anonymity of voters: The identity of the voter is untraceable.
- Accuracy of votes: Everyone can verify the validity of the votes and assure that the votes cannot be altered, duplicated, and forged by any others (including the authority).
- Collision freedom: To guarantee that all legal votes differ from each other.
- Tally correctness: The final tally must be equal to the total number of the valid votes.
- Verifiability: Each voter can check if his/she own vote is published and tallied.
- Double voting detection: If double voting occurs, the authority can find who the voter is.

However, these schemes [6,9,12] require such cryptographic techniques as the blind signature [1,3,4] or some anonymous communication channels [2,5,16]. According to Karro and Wang's study [13], anonymous schemes have some weaknesses. One of the weaknesses is that if two or more voters happen to choose the same keys, then these originally valid votes will be the same. It does not meet the requirement of collision freedom, and hence, the tally center would decide that they are fraudulent votes and cross them out. Another weakness is that the anonymous channels are hard to implement over the Internet because the users would then be forced to browse in certain sites.

Mu and Varadharajan [15] have recently proposed an anonymous, secure electronic voting scheme which satisfies all the requirements for electronic voting. Mu and Varadharajan's scheme is based on the ElGamal digital signature algorithm and is suitable for large-scale elections. The authors claim that the system can detect the identities of the voters who send in wrongful votes on the condition of double voting. So far, many techniques [6,9,10,12,14,17] have been proposed to perform electronic voting, but they cannot detect the illegal voters. Only Mu and Varadharajan's scheme can accomplish this mission. However, we shall show that Mu and Varadharajan's scheme has a deadly weakness in security that legal voters can actually successfully vote more than once without being detected.

In this paper, we shall show the weakness of Mu and Varadharajan's scheme and propose an improved version to overcome the weakness and also to satisfy all the requirements for electronic voting. The organization of the paper is as follows. In the next section, we shall briefly review Mu and Varadharajan's scheme and show the weakness of their scheme. In Section 3, we shall propose a new secure anonymous electronic voting scheme and analyze the security of the improved scheme. Finally, we shall summarize the benefits of our scheme in the last section of this paper.

2. Mu and Varadharajan's anonymous electronic voting scheme and its weakness

Mu and Varadharajan [15] have proposed two versions of an anonymous secure electronic voting scheme to be applied over the network. Both versions are based on the ElGamal digital signature [11] without any anonymous channel. One of the two assumes that the Authentication Server (AS) is trusted, and therefore it does not generate any voting ticket without the voter's consent. In addition, it does not leak out any information to the Voting Servers (VS) or Ticket Counting Server (TCS). The other version assumes that the AS is not to be trusted, which is closer to the truth. We shall emphasize the version that assumes the AS is not to be trusted.

2.1. Mu and Varadharajan's anonymous secure electronic voting scheme

In Mu and Varadharajan's voting scheme, participants are the following parties: Voters, an Authentication Server, Voting Servers, a Ticket Counting Server, and a Certificate Authentic. The scheme is composed of three phases: the voting ticket obtaining phase, the voting and tickets collecting phase, and the tickets counting phase.

For the rest of this paper, V will denote the Voter, AS the Authentication Server, VS the Voting Servers, TCS the Ticket Counting Server, CA the Certificate Authentic, t the timestamp, p a large prime number, and $\|$ a concatenation of bits. Before performing the voting processes, each participant x has a long-term RSA [18] key pair $\{e_x, d_x\}$ and a product of two large prime numbers n_x, where $e_x \times d_x \bmod \phi(n_x) = 1$. For

example, a voter V holds an RSA key pair $\{e_V, d_V\}$ and n_V; AS holds an RSA key pair $\{e_{AS}, d_{AS}\}$ and n_{AS}. Furthermore, the eligible voters hold a long-term voting certificate (Cert) delivered from CA. The certificate is signed by the CA's secret key, and the contents include a serial number, the voter's identity, the CA's identity, the RSA public key, the life-cycle time, and a timestamp.

2.1.1. Step 1. The voting ticket obtaining phase

1. Before sending Cert to prove that he/she is an eligible voter, a voter first chooses a blind factor b and three random numbers $g, r, k_1 \in Z_p^*$, where Z_p^* is the set of all positive integers smaller than and relatively prime to p. Then, the voter computes the parameters a, x_1, x_2, and x_3 by using the following equations:

$$a = g^r \bmod p,$$

$$x_1 = gb^{e_{AS}} \bmod n_{AS}, \quad (1)$$

$$x_2 = g^{k_1} b^{e_{AS}} \bmod n_{AS}, \text{ and}$$

$$x_3 = ab^{e_{AS}} \bmod n_{AS}. \quad (2)$$

   The voter sends $\{V, AS, Cert_V, (x_1 \| x_2 \| x_3 \| t)^{d_V} \bmod n_V\}$ to AS.
2. AS first verifies the validity of the certificate and validates the signature $(x_1 \| x_2 \| x_3 \| t)^{d_V} \bmod n_V$. Then AS chooses a random number k_2 and computes:

$$x_4 = (k_2 \| t)^{e_V} \bmod n_V,$$
$$x_5 = (x_1^{3k_2} x_2^2 x_3)^{d_{AS}} \bmod n_{AS}$$
$$= (y_1 y_2 a)^{d_{AS}} b^{3(k_2+1)} \bmod n_{AS},$$

   where $y_1 = g^{k_1 + k_2}$ and $y_2 = g^{k_1 + 2k_2}$. The messages $\{AS, V, x_4, (x_5 \| t)^{e_V} \bmod n_V\}$ are delivered to V. Note that the parameter k_2 is different for each voter and that AS stores k_2 along with V's identity in its database.
3. V obtains k_2 by decrypting x_4. Thus, V can calculate y_1 and y_2. Furthermore, V can derive s by removing the blind factor $b^{3(k_2+1)}$ following the equation below:

$$s = x_5 b^{-3(k_2+1)} = (y_1 y_2 a)^{d_{AS}} \bmod n_{AS}.$$

   s is the signature for the product $y_1 y_2 a$. The signatures s_1 and s_2 of the voting content m can be generated only from V by the ElGamal digital signature algorithm [11]:

$$s_1 = (k_1 + k_2)^{-1}(ma - r) \bmod p - 1, \text{ and} \quad (3)$$

$$s_2 = (k_1 + 2k_2)^{-1}(ma - r) \bmod p - 1. \quad (4)$$

   Then V can obtain the voting ticket $T = \{a \| g \| y_1 \| y_2 \| s \| s_1 \| s_2 \| m\}$.

2.1.2. Step 2. The voting and tickets collecting phase

After obtaining a valid voting ticket from AS, V can cast the vote to the Voting Server over the network. The main task of VS is to guarantee the validity of the voting ticket.

1. V sends the voting ticket T to VS.
2. VS decrypts T and verifies the validity of a, y_1, y_2 by AS's signature s. Then, VS checks the correctness of the signatures s_1 and s_2 on m by the following equations:

$$y_1^{s_1} a = g^{ma} \bmod p, \text{ and} \quad (5)$$

$$y_2^{s_2} a = g^{ma} \bmod p. \quad (6)$$

If the result of the above verification is positive, then VS can make sure the ticket T is valid. VS stores all the voting tickets cast in and sends this batch to the Ticket Counting Server over the network.

2.1.3. Step 3. The tickets counting phase

All the Voting Servers send their tickets to the Ticket Counting Server. However, V may use the same parameters a, g, k_1, k_2 to sign another voting content m' and send the new ticket $T = \{a \| g \| y_1 \| y_2 \|$

$s \| s_1' \| s_2' \| m'\}$ to a different VS. For double voting detection, VS checks the part parameters a, g, y_1, y_2 of T to see whether they have been repetitively used. If the parameters have been used more than once, it means there is a case of double voting. VS has the ability to find who the voter is by the following equations:

$$k_1 + k_2 = \frac{m'a - ma}{s_1' - s_1} \bmod (p-1), \text{ and} \quad (7)$$

$$k_1 + 2k_2 = \frac{m'a - ma}{s_2' - s_2} \bmod (p-1). \quad (8)$$

From Eqs. (7) and (8), VS can obtain the parameter k_2, and hence, VS can find the identity of the illegal voter. In this phase, the main tasks of VS are to count the tickets and to prevent any voter from casting more than once in different voting servers.

2.2. The weakness of Mu and Varadharajan's scheme

In this subsection, we shall show that Mu and Varadharajan's scheme has a weakness in security that a voter can in fact vote more than once without being detected. The attacks are described as follows.

2.2.1. Attack 1

In Step 1, voter V can obtain a valid ticket $T=\{a \| g \| y_1 \| y_2 \| s \| s_1 \| s_2 \| m\}$. The voter can successfully vote more than once in the following process. First, V computes g', y_1', y_2', a' as

$$g' = g^{c_0} \bmod p,$$

$$y_1' = g'^{(k_1+k_2+c_1)c_0^{-1}} \bmod p,$$

$$y_2' = g'^{(k_1+2k_2+c_2)c_0^{-1}} \bmod p,$$

$$a' = g'^{(r+c_3)c_0^{-1}} \bmod p,$$

where c_0, c_1, c_2, and c_3 are integers, $c_1, c_2, c_3 \neq 0$ and $c_1 + c_2 + c_3 = 0$. Therefore, V can generate a new ticket $T'=\{a' \| g' \| y_1' \| y_2' \| s \| s_1' \| s_2' \| m\}$, where s_1' and s_2' are the signatures of m created with the keys $(k_1+k_2+c_1)c_0^{-1}$ and $(k_1+2k_2+c_2)c_0^{-1}$, respectively.

$$s_1' = ((k_1+k_2+c_1)c_0^{-1})^{-1}(ma' - (r+c_3)c_0^{-1}) \bmod p-1, \text{ and}$$

$$s_2' = ((k_1+2k_2+c_2)c_0^{-1})^{-1}(ma' - (r+c_3)c_0^{-1}) \bmod p-1.$$

In the voting and tickets collecting phase, V can send the new ticket T' to VS. VS first verifies signature s and checks the validity of a', y_1', y_2' following the equation:

$$s^{e_{AS}} = y_1 y_2 a \bmod n_{AS} = y_1'\ y_2'\ a' \bmod n_{AS}.$$

VS then verifies the validity of s_1' and s_2' following Eqs. (5) and (6). Thus, VS believes the ticket T' is valid and sends it to TCS. For double voting protection, TCS checks the parameters a', g', y_1', y_2' and decides that they have not been used more than once. Thus, the attack can succeed without being detected. Even if VS detects that the signature s has in fact been used before, it still cannot detect the identity of the illegal voter by Eqs. (7) and (8).

2.2.2. Attack 2

Similar to Attack 1, V first chooses a random number h and computes $g' = g^h$, $a' = a^{h^2}$, $y_1' = y_1^{h^2}$, $y_2' = y_2^{h^2}$, and $s' = s^{h^2}$. Furthermore, the signatures s_1' and s_2' of m can be computed by Eqs. (3) and (4), respectively.

$$s_1' = (k_1+k_2)^{-1}h^{-1}(ma' - hr) \bmod p-1, \text{ and}$$

$$s_2' = (k_1+2k_2)^{-1}h^{-1}(ma' - hr) \bmod p-1.$$

Therefore, V can generate a new voting ticket $T'=\{a' \| g' \| y_1' \| y_2' \| s' \| s_1' \| s_2' \| m\}$, where s_1' and s_2' are computed by Eqs. (3) and (4) using the keys $(k_1+k_2)h$ and $(k_1+2k_2)h$, respectively. VS then

believes T' is a valid ticket because the following verifications hold.

$$(s')^{e_{AS}} = (y_1 y_2 a)^{h^2} = y_1' y_2' a' \bmod n_{AS},$$

$$y_1'^{s_1'} a' = g'^{ma'} \bmod p,$$

$$y_2'^{s_2'} a' = g'^{ma'} \bmod p.$$

Using the above two attacks, a voter can vote more than once and can remain undetected.

3. The improvement on Mu and Varadharajan's anonymous electronic voting scheme

In this section, we shall propose a secure electronic voting scheme for enhancing the security of Mu and Varadharajan's scheme, and we shall also analyze the security of the proposed scheme to prove its security. This new scheme can overcome the weakness of Mu and Varadharajan's scheme as described in Section 2.2.

3.1. The improved scheme

The proposed scheme is also composed of three phases with the some notations as are described in Section 2.1. The details of the improved scheme are as follows:

3.1.1. Step 1. The voting ticket obtaining phase

1. A voter chooses two blind factors b_1 and b_2 as well as two random numbers k_1 and r. With these parameters, the voting system computes w_1 and w_2 by using the following equations:

$$w_1 = g^r b_1^{e_{AS}} \bmod n_{AS}, \text{ and}$$

$$w_2 = g^k b_2^{e_{AS}} \bmod n_{AS}.$$

where $g \in Z_p^*$ is the system's public parameter. Then, the voter sends {V, AS, Cert_V, t, w_1, w_2, $((w_1 \| w_2 \| t)^{d_V} \bmod n_V)$} to AS.

2. AS first verifies the validity of the certificate and validates the signature $(w_1 \| w_2 \| t)^{d_V} \bmod n_V$. If the verification result is positive, AS can make sure that the received parameters are correct. Then AS chooses a random number k_2 is different for each voter and computes:

$$w_3 = (k_2 \| t)^{e_V} \bmod n_V$$

$$w_4 = (w_1 \times \text{AS})^{d_{AS}} \bmod n_{AS} = (a \times \text{AS})^{d_{AS}} b_1 \bmod n_{AS},$$

$$w_5 = (w_2 \times g^{k_2} \times \text{AS})^{d_{AS}} \bmod n_{AS} = (y_1 \times \text{AS})^{d_{AS}} b_2 \bmod n_{AS},$$

$$w_6 = (w_2^2 \times g^{k_2} \times \text{AS})^{d_{AS}} \bmod n_{AS} = (y_2 \times \text{AS})^{d_{AS}} b_2^2 \bmod n_{AS},$$

where $a = g^r$, $y_1 = g^{k_1 + k_2}$, and $y_2 = g^{2k_1 + k_2}$. The messages {AS, V, w_3, $(w_4 \| w_5 \| w_6 \| t)^{e_V} \bmod n_V$} are delivered to V. Note that AS stores k_2 along with V's identity in its database.

3. V obtains k_2 by decrypting w_3. Thus, V can calculate y_1 and y_2. Furthermore, V can calculate the signatures s_1, s_2, and s_3 by removing the blind factors following the equations below:

$$s_1 = w_4 \times b_1^{-1} = (a \times \text{AS})^{d_{AS}} \bmod n_{AS},$$

$$s_2 = w_5 \times b_2^{-1} = (y_1 \times \text{AS})^{d_{AS}} \bmod n_{AS},$$

$$s_3 = w_6 \times b_2^{-2} = (y_2 \times \text{AS})^{d_{AS}} \bmod n_{AS}.$$

4. The voter applies the ElGamal digital signature scheme to sign the voting content m. Let y_1 and y_2 be the public keys of the ElGamal Cryptosystem,

and $x_1 = k_1 + k_2$ and $x_2 = 2k_1 + k_2$ be the corresponding secret keys, such that $y_1 = g^{k_1 + k_2} \bmod p$ and $y_2 = g^{2k_1 + k_2} \bmod p$. The two signatures (a, s_4) and (a, s_5) of the voting content m can be generated from the following equations

$$s_4 = x_1^{-1}(ma - r) \bmod p - 1, \text{ and}$$

$$s_5 = x_2^{-1}(ma - r) \bmod p - 1,$$

respectively. Then V can obtain the voting ticket as $T = \{s_1 \| s_2 \| s_3 \| s_4 \| s_5 \| a \| y_1 \| y_2 \| m\}$.

3.1.2. Step 2. The voting and tickets collecting phase

1. V sends the voting ticket T to VS.
2. VS verifies the validity of a, y_1, and y_2 by checking the following equations:

$$\mathrm{AS} \times a \stackrel{?}{=} s_1^{e_{\mathrm{AS}}} \bmod n_{\mathrm{AS}}, \quad (9)$$

$$\mathrm{AS} \times y_1 \stackrel{?}{=} s_2^{e_{\mathrm{AS}}} \bmod n_{\mathrm{AS}}, \text{ and} \quad (10)$$

$$\mathrm{AS} \times y_2 \stackrel{?}{=} s_3^{e_{\mathrm{AS}}} \bmod n_{\mathrm{AS}}. \quad (11)$$

If all are positive, VS also verifies the correctness of the signatures (a, s_4) and (a, s_5) on m by checking the following equations:

$$g^{ma} \stackrel{?}{=} y_1^{s_4} a = \bmod p, \text{ and}$$

$$g^{ma} \stackrel{?}{=} y_2^{s_5} a = \bmod p,$$

respectively. If both the verifications turn out positive, VS can make sure the ticket T is valid. VS stores all the voting tickets cast in and sends this batch to TCS over the network.

3.1.3. Step 3. The tickets counting phase

All the Voting Servers send their tickets to TCS. TCS publishes the tickets and counts them. Besides that, it is also responsible for detecting double voting. The double voting detection process is as follows.

1. Assume that a voter uses the same parameters y_1, y_2, and a to sign another voting content m' and sends the new voting ticket to a different VS.
2. For double voting detection, TCS first checks y_1, y_2, and a of T to see whether they have been used before.
3. If the parameters have been used more than once and the voting contents are not the same, it means that there is a case of double voting. TCS has the ability to find who the voter is by figuring out the following equations:

$$x_1 = \frac{m'a - ma}{s_4' - s_4} \bmod (p - 1), \text{ and}$$

$$x_2 = \frac{m'a - ma}{s_5' - s_5} \bmod (p - 1).$$

From the above two equations, TCS can obtain the parameters from computing $x_2 - x_1 = (2k_1 + k_2) - (k_1 + k_2) = k_1$. Consequently, k_2 can be easily recovered by computing $x_1 - k_1 = k_2$, and hence, TCS can find the identity of the illegal voter. Since the contents of votes are open, the complexity problem in counting votes in large-scale systems is overcome in these systems.

The flowchart of the improved scheme is in Fig. 1.

3.2. The security of the improved scheme

Our improved scheme is also based on the ElGamal digital signature and satisfies all the electronic voting requirements. The proposed scheme enhances the security and overcomes the weakness of Mu and Varadharajan's scheme. In the following analysis, we shall show that our scheme can resist the previous attacks and the parties conspiring attack.

3.2.1. Resisting the previous attacks

Assume that a voter wants to forge the parameters a, y_1, and y_2 like what happens in Attack 1. However, it is impossible to make corresponding signatures s_1,

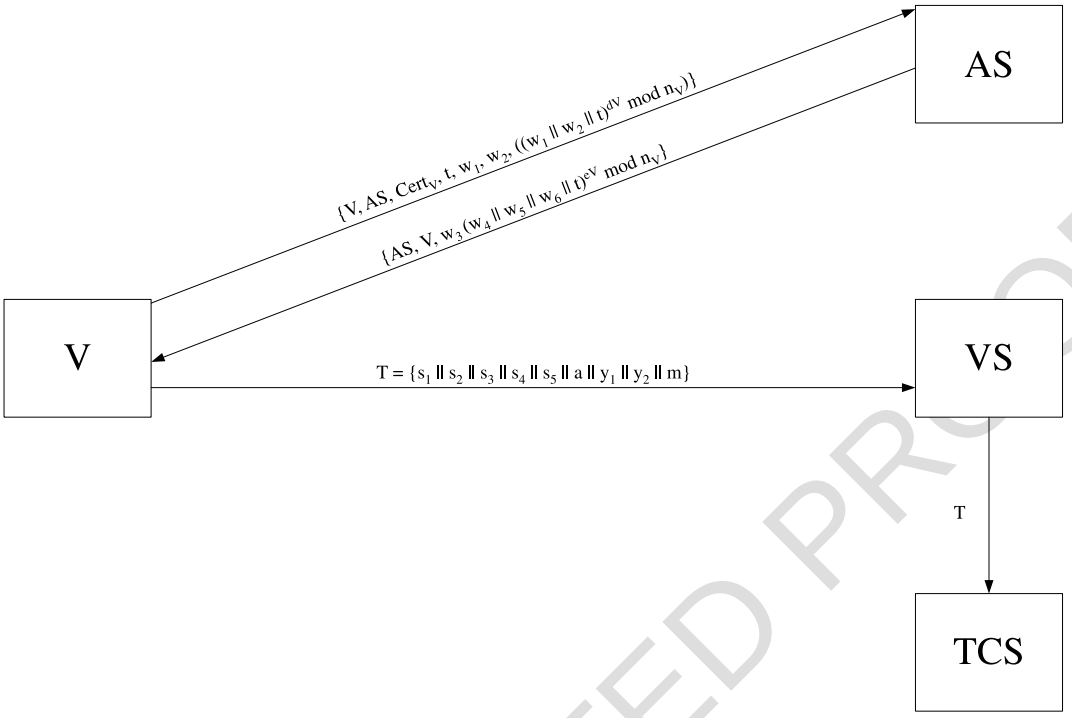

Fig. 1. The flowchart of the improved scheme.

s_2, and s_3 that satisfy Eqs. (9), (10), and (11), respectively. The voter does not know the AS's private key d_{AS}. Hence, these forged parameters still cannot escape the verifications.

In the second attack, any voter can also make a forged signature $s_2'=(AS'\times y_2')^{d_{AS}}$ easily, but he/she cannot generate the correct signatures on the voting content m. For instance, suppose $s_2' = s_2^2=(AS^2 \times y_2^2)^{d_{AS}}$ and $y_2' = AS \times y_2^2$. Hence, the parameter y_2' can pass the verification. However, the voter cannot obtain the corresponding secret key x_1' due to the difficulty of computing discrete logarithms. Without the secret key, the voter cannot generate the correct signature. Therefore, Attack 2 cannot do any harm to the improved scheme. A voter can never generate another valid vote.

3.2.2. Resisting the parties conspiring attack

First, we assume that two or more voters conspire to obtain a new signature from individual signatures. Suppose that two voters V_1 and V_2 have valid signatures (s_{11}, s_{12}, s_{13}) and (s_{21}, s_{22}, s_{23}), respectively. They conspire to obtain a new signature of (s_1', s_2', s_3') by the following equation:

$$s_i' = s_{1i} \times s_{2i} \bmod n_{AS}$$

where $i=1, 2,\ldots, 4$. However, to compute the corresponding parameters r', x_1', and x_2', the voters have to solve discrete logarithms.

On the other hand, we assume that AS, VS, and TCS are not to be trusted in our scheme. AS may leak out the request information to VS or TCS. Even if AS, VS, and TCS have all the voting information, they still cannot trace the identities of the voters. In addition, if AS, VS, and TCS try to conspire to forge a valid voting ticket, they still cannot do any harm to this scheme. The reason is that T is public, which means everyone can check whether the total number of voting tickets is equal to the total number of registered voters. Thus, our scheme can prevent the parties conspiring attack.

Furthermore, a voter may try to use the same secret key to sign a different content m'. However, the scheme also has the ability to detect who is using the same

secret key to double vote as described in the tickets counting phase. Therefore, the improved scheme can prevent a voter from voting more than once.

4. Conclusion

In this article, we have proposed a secure electronic voting scheme that overcomes the weakness of Mu and Varadharajan's scheme. In addition, the scheme also meets all the electronic voting requirements: anonymity of voters, accuracy of voters, collision freedom, tally correctness, verifiability, and double voting detection. So far, there are few methods to meet all the electronic voting requirements, especially double voting detection. Moreover, the scheme is suitable for large-scale elections and does not require any special voting channel.

References

[1] J. Camenisch, J. Piveteau, M. Stadler, Blind signatures based on discrete logarithm problem, Advances in Cryptology, EUROCRYPT'94, Lecture Notes in Computer Science 950 (1994) 428–432.
[2] D. Chaum, Untracable electronic mail, return addresses, and digital pseudonyms, Communication of the ACM 24 (1981) 84–88.
[3] D. Chaum, Blind signatures for untraceable payments, Advances in Cryptology, CRYPTO'82 (1982) 199–203.
[4] D. Chaum, Blind signatures system, Advances in Cryptology, CRYPTO'83 (1983) 153–156.
[5] D. Chaum, The dinning cryptographers problem: unconditional sender and recipient untraceability, Journal of Cryptography 1 (1988) 65–75.
[6] D. Chaum, Elections with unconditionally secret ballots and disruption equivalent to breaking RSA, Advances in Cryptology, EUROCRYPT'88 (1988) 177–182.
[7] R. Cramer, M. Franklin, B. Schoenmakers, M. Yung, Multi-authority secret ballot elections with linear work, Advances in Cryptology, EUROCRYPT'96, Lecture Notes in Computer Science 1070 (1996) 72–83.
[8] R. Cramer, R. Gennaro, J. Borrell, A secure and optimally efficient multi-authority election scheme, Advances in Cryptology, EUROCRYPT'97, Lecture Notes in Computer Science 1233 (1997) 103–117.
[9] L. Cranor, R. Cytron, Sensus: a security-conscious electronic polling system for the Internet, Proceedings of the International Conference on System Sciences, Hawaii, 1997.
[10] G. Dini, Electronic voting in a large-scale distributed system, Networks 38 (2001) 22–32.
[11] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory IT-31 (1985 (July)) 469–472.
[12] A. Fujioka, T. Okamoto, K. Ohta, A practical secret voting scheme for large-scale elections, Advances in Cryptology, AUSCRYPT'92, Lecture Notes in Computer Science 718 (1993) 244–251.
[13] J. Karro, J. Wang, Towards a practical, secure, and very large scale online election, Proceedings of the 15th Annual Computer Security Applications Conference, CACSAC'99, 1999, pp. 161–169.
[14] C.L. Lei, C.I. Fan, A universal single-authority election system, IEICE Transactions on Fundamentals E81-A (10) (1998) 2186–2193.
[15] Y. Mu, V. Varadharajan, Anonymous secure e-voting over a network, Proceedings of the 14th Annual Computer Security Applications Conference, CACSAC'98, 1998, pp. 299–2936.
[16] A. Pfitzmann, A switched/broadcase ISDN to decrease user oberability, Proceedings of the International Zurich Seminar on Digital Communication, 1984, pp. 183–190.
[17] I. Ray, N. Narasimhamurthi, An anonymous electronic voting protocol for voting over the internet, Proceedings of Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, 2001, pp. 188–190.
[18] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM 21 (1978 (Feb.)) 120–126.

I.-C. Lin received his BS in Computer and Information Sciences from Tung Hai University, Taichung, Taiwan, Republic of China, in 1998 and his MS in Information Management from Chaoyang University of Technology, Taiwan, in 2000. He is currently pursuing his PhD degree in Computer Science and Information Engineering from the National Chung Cheng University. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

M.-S. Hwang received his BS in Electronic Engineering from the National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; his MS in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988; and his PhD in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at the National Cheng Kung University, Taiwan, from 1984 to 1986. Dr. Hwang passed the National Higher Examination infield "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering," qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997,1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is currently a professor and chairman of the Department of Information Management, Chaoyang University of Technology, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications.

Chin-Chen Chang was born in Taichung, Taiwan, the Republic of China, on November 12, 1954. He received his BS degree in Applied Mathematics in 1977 and his MS degree in Computer and Decision Sciences in 1979 from the National Tsing Hua University, Hsinchu, Taiwan. He received his PhD in Computer Engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. From 1983 to 1989, he was among the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. Since August 1989, he has worked as a professor of the Institute of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan. Dr. Chang is a Fellow of IEEE and a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, and the Phi Tau Phi Society of the Republic of China. His research interests include computer cryptography, data engineering, and image compression.