

## Double Voting Problem of Some Anonymous E-Voting Schemes

MAHDI ASADPOUR AND RASOOL JALILI

*Department of Computer Engineering*

*Sharif University of Technology*

*Tehran, Iran*

*E-mail: {asadpur@ce.; jalili@}sharif.edu*

In 1998, Mu and Varadharajan proposed an anonymous e-voting scheme to be applied in both small-scale and large-scale elections. They claimed that it protects the anonymity of voters and prevents double voting. They also mentioned that if a malicious voter votes more than once, his identity will be discovered by the election authority. In 2003, Lin *et al.* demonstrated that the scheme fails to resist double voting and further proposed an enhanced scheme to overcome the weakness. Their scheme has received a much of interest by Yang *et al.* (2004), Hwang *et al.* (2005), and Rodriguez-Henriquez *et al.* (2007). They have improved Lin *et al.*'s scheme to prevent their proposed weaknesses. This paper indicates that in these anonymous e-voting schemes (started from Lin *et al.*'s scheme), a valid voter in cooperation with another valid voter can successfully vote more than once. It is also demonstrated that the scheme is unable to recognize a malicious voter who has double voted.

**Keywords:** e-voting, double voting, anonymity, cryptanalysis, blind signature

### 1. INTRODUCTION

An electronic voting (e-voting) system enables voters to perform voting over computer networks. It can be realized in a way that is more convenient, faster and cheaper than a conventional voting.

Security and privacy are always the crucial factors in e-voting systems. A secure e-voting scheme should fulfill several requirements [8, 10, 13] such as: only eligible voters are permitted to vote (eligibility), the privacy of voters is protected (anonymity), each voter can only vote once (so-called "double voting" prevention), and no receipt is provided which may be used for purchasing votes (receipt-freeness). A number of research publications have addressed this problem proposing secure e-voting systems [1, 8-11].

In 1998, Mu and Varadharajan [1] started a research track in this area by proposing two secure e-voting systems. Their schemes are based on ElGamal digital signature [12], and Chaum blind signature [2]. They claimed that, besides satisfying the other requirements, they can identify the malicious voter who makes double voting. In one of their schemes, it is assumed that the election authority includes a trusted authentication server which would not illegally vote on behalf of a voter. The other scheme assumes that there is no such a trusted entity, therefore it is more practical and we will focus on it.

In 2003, Lin *et al.* [4] pointed out that Mu and Varadharajan's scheme cannot prevent double voting and proposed an enhanced scheme to solve this problem. Yang *et al.* [5], 2004, proposed another improvement to Mu and Varadharajan' scheme, but it cannot

find out the identity of a malicious double voter. In 2005, Hwang *et al.* [6] mentioned that Lin *et al.*'s scheme suffers from another weakness inherited from the original as reported by [3]; in Lin *et al.*'s scheme the authority has the ability to identify the owner of a cast ticket, so the anonymity of voters is not provided. Hwang *et al.* further proposed an improved scheme to satisfy the anonymity requirement. Recently (2007), Rodriguez-Henriquez *et al.* [7] find that Lin *et al.*'s scheme suffers from an attack by a corrupted authentication server (however Hwang *et al.* solved it), and also from unfeasibility of signing a vote in some circumstances. They further make changes to Lin *et al.*'s scheme to prevent such weaknesses.

In this paper, we illustrate that in anonymous e-voting schemes which are based on Lin *et al.*'s scheme, a valid voter in cooperation with another valid voter can vote more than once while this multiple voting is not discovered. We also demonstrate that they fail to reveal the identity of a vicious voter who has voted twice even if double voting is detected. For the sake of brevity, we just describe these problems and a solution on Hwang *et al.*'s scheme. Similarly our descriptions can be adapted to the others.

The rest of this paper is organized as follows. In the following section we review the Hwang *et al.*'s approach. Section 3 describes how a legal voter is able to deceive the election authority and perform multiple voting, followed by a clue to solve this problem. Section 4 indicates the failure of the scheme in identifying a double voter. Finally, a concluding remark is given in section 5.

The cryptographic notations used throughout this paper are listed in Table 1.

**Table 1. Notations used throughout this paper.**

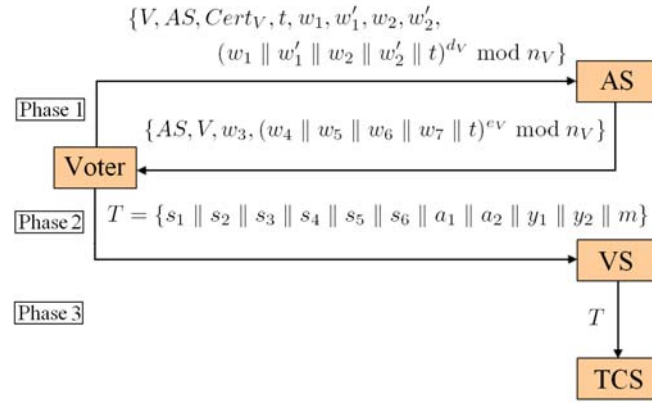
Symbols	Definition
$(e_x, n_x), d_x$	Public/Private key pair of participant $x$ .
$p$	A large prime number, which is public.
$g, h$	$g, h \in \mathbb{Z}_p^*$ are two public parameters of the system where $g \neq h$ .
$\parallel$	The concatenation operation.

## 2. HWANG ET AL.'S SCHEME

Hwang *et al.*'s e-voting system involves the following parties:

- A Certificate Authority (CA) that issues long term digital certificates to be used for a number of elections.
- Voters (V).
- An Authentication Server (AS) which authenticates voters and grants voting tickets.
- Some Voting Servers (VS) that are responsible for collecting tickets from voters.
- A Ticket Counting Server (TCS).

Hwang *et al.*'s e-voting scheme works in three phases: *the voting ticket obtaining phase*, *the voting and tickets collecting phase*, and *the tickets counting phase*. These phases are described as follows (summarized in Fig. 1).

Fig. 1. Three phases of Hwang *et al.*'s scheme.

## 2.1 The Voting Ticket Obtaining Phase

In this phase, each voter registers for the election and obtains a valid voting ticket from AS:

1. The voter  $V$  chooses two blind factors  $b_1$  and  $b_2$  as well as two random numbers  $k_1$  and  $r$ . Then,  $V$  computes  $w_1$ ,  $w'_1$ ,  $w_2$ , and  $w'_2$  through the following equations:

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

$$w'_1 = h^r b_1^{e_{AS}} \bmod n_{AS},$$

$$w_2 = g^{k_1} b_2^{e_{AS}} \bmod n_{AS},$$

$$w'_2 = h^{k_1} b_2^{e_{AS}} \bmod n_{AS}.$$

Next,  $V$  sends a voting request  $\{V, AS, Cert_V, t, w_1, w'_1, w_2, w'_2, (w_1 || w'_1 || w_2 || w'_2 || t)^d_V \bmod n_V\}$  to AS, where  $V$  and AS respectively denote the identities of  $V$  and AS,  $Cert_V$  represents the digital certificate of  $V$  signed by CA, and  $t$  is the current timestamp.

2. Upon receipt of the request, AS checks the validity of  $t$  and legitimacy of  $Cert_V$ , next it uses the public key of  $V$  ( $e_V, n_V$ ) extracted from  $Cert_V$  to verify the signature  $(w_1 || w'_1 || w_2 || w'_2 || t)^d_V \bmod n_V$ . Then, a unique random number  $k_2$  is chosen and finally,  $w_3$  (as encrypted  $k_2$ ),  $w_4$ ,  $w_5$ ,  $w_6$  and  $w_7$  (as blind signatures) are generated:

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

$$\begin{aligned} w_4 &= (w_1 \times AS)^{d_{AS}} \bmod n_{AS} \\ &= (a_1 \times AS)^{d_{AS}} \times b_1 \bmod n_{AS}, \end{aligned}$$

$$\begin{aligned} w_5 &= (w'_1 \times AS)^{d_{AS}} \bmod n_{AS} \\ &= (a_2 \times AS)^{d_{AS}} \times b_1 \bmod n_{AS}, \end{aligned}$$

$$\begin{aligned}
w_6 &= (w_2 \times g^{k_2} \times AS)^{d_{AS}} \bmod n_{AS} \\
&= (y_1 \times AS)^{d_{AS}} \times b_2 \bmod n_{AS}, \\
w_7 &= ((w_2')^2 \times h^{k_2} \times AS)^{d_{AS}} \bmod n_{AS} \\
&= (y_2 \times AS)^{d_{AS}} \times b_2^2 \bmod n_{AS},
\end{aligned}$$

where it is assumed that  $a_1 = g^r$ ,  $a_2 = h^r$ ,  $y_1 = g^{k_1+k_2}$  and  $y_2 = h^{2k_1+k_2}$ .  $AS$  delivers the message

$$\{AS, V, w_3, (w_4 \parallel w_5 \parallel w_6 \parallel w_7 \parallel t)^{e_V} \bmod n_V\}$$

to  $V$ . The unique  $k_2$  along with  $V$ 's identity is also recorded in the database of  $AS$ .

- Using the private key  $d_V$ ,  $V$  decrypts  $w_3$  to obtain  $k_2$ . Then  $V$  calculates  $y_1$  and  $y_2$ , the public keys of the ElGamal cryptosystems by utilizing  $g$ ,  $h$ ,  $k_1$  and  $k_2$ . Furthermore, it removes the blind factors to compute the signatures  $s_1$ ,  $s_2$ ,  $s_3$  and  $s_4$  as:

$$\begin{aligned}
s_1 &= w_4 \times b_1^{-1} = (a_1 \times AS)^{d_{AS}} \bmod n_{AS}, \\
s_2 &= w_5 \times b_1^{-1} = (a_2 \times AS)^{d_{AS}} \bmod n_{AS}, \\
s_3 &= w_6 \times b_2^{-1} = (y_1 \times AS)^{d_{AS}} \bmod n_{AS}, \\
s_4 &= w_7 \times b_2^{-2} = (y_2 \times AS)^{d_{AS}} \bmod n_{AS}.
\end{aligned}$$

- Constructing a voting ticket,  $V$  selects the candidate(s) to whom he wishes to vote for. Suppose the ballot  $m$  is created in a standard format followed by all voters.  $V$  applies the ElGamal digital signature scheme to generate two signatures  $(a_1, s_5)$  and  $(a_2, s_6)$  of the voting content  $m$ :

$$\begin{aligned}
s_5 &= x_1^{-1}(ma_1 - r) \bmod p-1, \\
s_6 &= x_2^{-1}(ma_2 - r) \bmod p-1.
\end{aligned}$$

Here,  $x_1 = k_1 + k_2$  and  $x_2 = 2k_1 + k_2$  stand for the private keys of the ElGamal cryptosystem. The voting ticket  $T$  is now constructed as:

$$T = \{s_1 \parallel s_2 \parallel s_3 \parallel s_4 \parallel s_5 \parallel s_6 \parallel a_1 \parallel a_2 \parallel y_1 \parallel y_2 \parallel m\}.$$

## 2.2 The Voting and Tickets Collecting Phase

In this phase, every registered voter can vote by sending his ticket to a voting server ( $VS$ ) over a public network. Validating the authenticity of the received tickets,  $VS$  stores acceptable tickets in its database. Finally, the collected tickets will be sent to the ticket counting server ( $TCS$ ), when the time of election is expired. The details are as:

1.  $V$  sends the voting ticket  $T$  to  $VS$ .
2. Upon receipt,  $VS$  verifies the validity of  $a_1$ ,  $a_2$ ,  $y_1$  and  $y_2$  by checking the equations:

$$AS \times a_1 = s_1^{e_{AS}} \pmod{n_{AS}},$$

$$AS \times a_2 = s_2^{e_{AS}} \pmod{n_{AS}},$$

$$AS \times y_1 = s_3^{e_{AS}} \pmod{n_{AS}},$$

$$AS \times y_2 = s_4^{e_{AS}} \pmod{n_{AS}}.$$

If all of the validations succeed,  $VS$  examines the signatures  $(a_1, s_5)$  and  $(a_2, s_6)$  of  $m$  through the ElGamal verification scheme:

$$g^{ma_1} = y_1^{s_5} \times a_1 \pmod{p},$$

$$h^{ma_2} = y_2^{s_6} \times a_2 \pmod{p}.$$

If above equations hold,  $T$  is stored as a genuine voting ticket in the database of  $VS$ .

3. At the end of this phase, when the election time expires,  $VS$  sends all the collected tickets to  $TCS$ .

### 2.3 The Tickets Counting Phase

In this phase the received tickets from  $VS$ s are counted by  $TCS$  in order to announce the results of the election.

Upon receiving all tickets,  $TCS$  checks for double voting and detects the tickets which have been used twice or more. For this purpose, the parameters  $(y_1, y_2, a_1, a_2)$  of every ticket are examined to see whether they have been repeatedly used. If these parameters appear in more than one ticket, a double voting is discovered. Otherwise, the content  $m$  of the balloted ticket is counted in favor of the corresponding candidate(s).

If double voting occurs and a voter uses the same parameters  $(y_1, y_2, a_1, a_2)$  to sign two different voting contents  $m$  and  $m'$ ,  $TCS$  and  $AS$  collaborate to identify the malicious voter. In this regard,  $TCS$  can calculate the ElGamal private keys  $(x_1, x_2)$  and subsequently retrieve the  $k_2$  parameter of this voter as:

$$x_1 = \frac{m'a_1 - ma_1}{s'_5 - s_5} \pmod{p-1},$$

$$x_2 = \frac{m'a_2 - ma_2}{s'_6 - s_6} \pmod{p-1},$$

$$k_1 = x_2 - x_1 = (2k_1 + k_2) - (k_1 + k_2),$$

$$k_2 = x_1 - k_1.$$

*TCS* sends the revealed  $k_2$  to *AS*. *AS* searches its database and finds out the associated voter of the unique number  $k_2$ .

### 3. THE POSSIBILITY OF DOUBLE VOTING

The first subsection proposes an attack scenario to illustrate how a valid voter can vote more than once by utilizing the signed values of another valid voter. We shall prove that such double voting is not detected in Hwang *et al.*'s scheme until the number of counted tickets exceeds the number of all registered voters at the end of election.

The second subsection brings a clue to eliminate such a double voting weakness.

#### 3.1 Proposed Attack

Theorem 2 of Mu and Varadharajan's e-voting scheme indicates that if two voters use the same value of  $g$  and different values of  $r$ , they can double vote by exchanging the signed values [1]. To remedy this problem, they have included another signature in each voting ticket that makes a link between all signed values in the ticket to be sure that these values have to be used all together at the same time.

In Hwang *et al.*'s scheme, the parameters  $g$  and  $h$ , playing the same role as  $g$  in Mu and Varadharajan's scheme, are common among all voting tickets. On the other hand, there is not any linking parameter in each ticket to connect the signed values. Thus double voting attack can be realized in their scheme. The attack scenario is described as follows (depicted in Fig. 2):

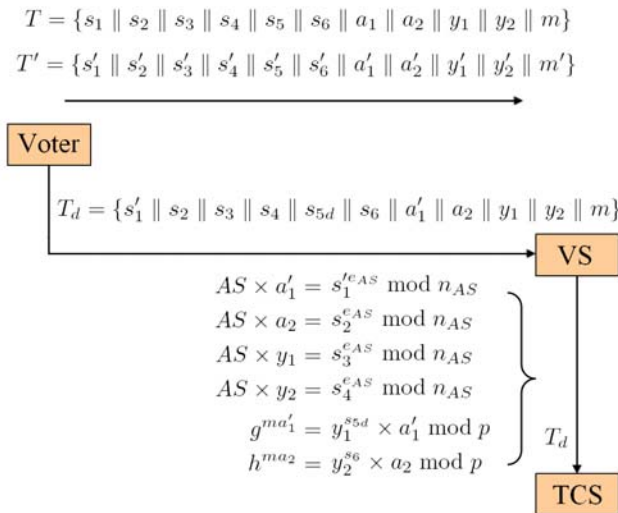


Fig. 2. Double voting attack on Hwang *et al.*'s scheme.

1. Suppose that voter  $V$  has voted with a valid ticket:

$$T = \{s_1 \parallel s_2 \parallel s_3 \parallel s_4 \parallel s_5 \parallel s_6 \parallel a_1 \parallel a_2 \parallel y_1 \parallel y_2 \parallel m\}$$

where  $a_1 = g^r \bmod p$ . Also suppose that voter  $V$  has voted with a valid ticket:

$$T' = \{s'_1 \parallel s'_2 \parallel s_3 \parallel s'_4 \parallel s'_5 \parallel s'_6 \parallel a'_1 \parallel a'_2 \parallel y'_1 \parallel y'_2 \parallel m'\}$$

where  $a'_1 = g^{r'} \bmod p$ .

2.  $V$  borrows  $r'$ ,  $a'_1$  and  $s'_1$  from voter  $V'$ .
3. Then  $V$  computes  $s_{5d}$  as:

$$s_{5d} = x_1^{-1}(ma'_1 - r') \bmod p - 1.$$

4. Finally he sends the following ticket  $T_d$  to  $VS$ :

$$T_d = \{s'_1 \parallel s_2 \parallel s_3 \parallel s_4 \parallel s_{5d} \parallel s_6 \parallel a'_1 \parallel a_2 \parallel y_1 \parallel y_2 \parallel m\}.$$

5. In the voting and tickets collecting phase and upon receiving ticket  $T_d$  from  $V$ ,  $VS$  validates this ticket through the following equations:

$$AS \times a'_1 \stackrel{?}{=} (s'_1)^{e_{AS}} \bmod n_{AS}, \quad (1)$$

$$AS \times a_2 \stackrel{?}{=} s_2^{e_{AS}} \bmod n_{AS}, \quad (2)$$

$$AS \times y_1 \stackrel{?}{=} s_3^{e_{AS}} \bmod n_{AS}, \quad (3)$$

$$AS \times y_2 \stackrel{?}{=} s_4^{e_{AS}} \bmod n_{AS}. \quad (4)$$

and

$$g^{ma'_1} \stackrel{?}{=} y_1^{s_{5d}} \times a'_1 \bmod p, \quad (5)$$

$$h^{ma_2} \stackrel{?}{=} y_2^{s_6} \times a_2 \bmod p. \quad (6)$$

These validations succeed because:

- Eq. (1) was validated when ticket  $T'$  was checked by  $VS$ .
- Eqs. (2)-(4) and (6) also passed the  $VS$ 's checking when it validated the ticket  $T$ .
- At last, Eq. (5) can be verified due to the tricky way we chose  $s_{5d}$ .

6. In the tickets counting phase,  $TCS$  examines  $(y_1, y_2, a'_1, a_2)$  parameters of  $T_d$  to detect double voting. But since that for  $T$  we have  $(y_1, y_2, a_1, a_2)$  and for  $T'$  we have  $(y'_1, y'_2, a'_1, a'_2)$ , no duplication is discovered. Therefore,  $T_d$  is considered as a valid ticket and is successfully counted by  $TCS$ .

In a similar way, two valid voters can exchange their  $a_2$  parameter (with or without exchanging  $a_1$ ) and still be able to vote. Accordingly by using this attack, a valid voter can vote more than once while the illegal votes are not detected as forged tickets.

### 3.2 Clue to a Solution

As mentioned before, the success of the double voting attack against Hwang *et al.*'s

scheme is due to the fact that there is no logical connection among the signed values of a voting ticket. It causes that someone could change one of the signatures needless to replace the other ones. In order to overcome the described attack, the linking term  $H_{lnk} = Hash(a_1 || a_2)$  can be introduced. (*Hash* is a one-way hash function.) We suggest integrating  $H_{lnk}$  into the signed values  $s_1, s_2, s_3$  and  $s_4$  in such a way that they (together with  $y_1$  and  $y_2$ ) all depend to  $a_1$  and  $a_2$ . Thus, none of these values can be altered separately. In the following, we more focus on our changes to Hwang *et al.*'s scheme, since our scheme is quite similar to theirs, and then prove that why  $H_{lnk}$  can provide the functionality of resisting against double voting.

In our improvement for Hwang *et al.*'s scheme,  $V$  chooses another blind factor  $b_0$  and computes  $w_0$  through the following equation during the voting ticket obtaining phase:

$$\begin{aligned} H_{lnk} &= Hash(g^r || h^r) = Hash(a_1 || a_2), \\ w_0 &= H_{lnk} b_0^{e_{AS}} \bmod n_{AS}. \end{aligned}$$

$w_1, w'_1, w_2$  and  $w'_2$ , are generated the same as before.  $V$  sends the modified voting request  $\{V, AS, Cert_V, t, w_0, w_1, w'_1, w_2, w'_2, (w_0 || w_1 || w'_1 || w_2 || w'_2 || t)^{d_V} \bmod n_V\}$  to  $AS$ .

$AS$  employs the received  $w_0$  in generation of the blind signatures ( $w_0$  instead of that static value  $AS$  in Hwang *et al.*'s scheme):

$$\begin{aligned} w_4 &= (w_1 \times w_0)^{d_{AS}} \bmod n_{AS} \\ &= (a_1 \times H_{lnk})^{d_{AS}} \times b_1 \times b_0 \bmod n_{AS}, \\ w_5 &= (w'_1 \times w_0)^{d_{AS}} \bmod n_{AS} \\ &= (a_2 \times H_{lnk})^{d_{AS}} \times b_1 \times b_0 \bmod n_{AS}, \\ w_6 &= (w_2 \times g^{k_2} \times w_0)^{d_{AS}} \bmod n_{AS} \\ &= (y_1 \times H_{lnk})^{d_{AS}} \times b_2 \times b_0 \bmod n_{AS}, \\ w_7 &= ((w'_2)^2 \times h^{k_2} \times w_0)^{d_{AS}} \bmod n_{AS} \\ &= (y_2 \times H_{lnk})^{d_{AS}} \times b_2^2 \times b_0 \bmod n_{AS}. \end{aligned}$$

$w_3$  is as before. In this way,  $H_{lnk}$  will be blindly integrated into signed values by  $AS$ .

After receiving the message from  $AS$  and doing some checking by  $V$ , it removes the blind factors ( $b_0, b_1, b_2$ ) and computes the signatures:

$$\begin{aligned} s_1 &= w_4 \times b_1^{-1} \times b_0^{-1} = (a_1 \times H_{lnk})^{d_{AS}} \bmod n_{AS}, \\ s_2 &= w_5 \times b_1^{-1} \times b_0^{-1} = (a_2 \times H_{lnk})^{d_{AS}} \bmod n_{AS}, \\ s_3 &= w_6 \times b_2^{-1} \times b_0^{-1} = (y_1 \times H_{lnk})^{d_{AS}} \bmod n_{AS}, \\ s_4 &= w_7 \times b_2^{-2} \times b_0^{-1} = (y_2 \times H_{lnk})^{d_{AS}} \bmod n_{AS}. \end{aligned}$$

$s_5$  and  $s_6$  will be generated through the same formula of Hwang *et al.*'s scheme.

Validating the signatures during the voting and tickets collecting phase,  $VS$  first computes  $H_{lnk} = Hash(a_1 || a_2)$ , then verifies the integrity of  $a_1, a_2, y_1$  and  $y_2$  by checking:

$$\begin{aligned}
H_{lnk} \times a_1 &= s_1^{e_{AS}} \pmod{n_{AS}}, \\
H_{lnk} \times a_2 &= s_2^{e_{AS}} \pmod{n_{AS}}, \\
H_{lnk} \times y_1 &= s_3^{e_{AS}} \pmod{n_{AS}}, \\
H_{lnk} \times y_2 &= s_4^{e_{AS}} \pmod{n_{AS}}.
\end{aligned}$$

Validity of the signatures  $(a_1, s_5)$  and  $(a_2, s_6)$  of  $m$  also are checked by VS similar to what is done in the base scheme. In the last phase, TCS examines the parameters  $(a_1, a_2, y_1, y_2)$  of every ticket to find out if they have been repeatedly used.

**Theorem 1** By integrating  $H_{lnk} = Hash(a_1 || a_2)$  in all of the signed values  $s_1, s_2, s_3$  and  $s_4$ , voters can not vote more than once by exchanging their parameters  $(a_1, a_2, y_1, y_2)$ .

**Proof:** Assume that voter  $V$  has voted with ticket  $T = \{s_1 || s_2 || s_3 || s_4 || s_5 || s_6 || a_1 || a_2 || y_1 || y_2 || m\}$  and voter  $V'$  has voted with ticket  $T' = \{s'_1 || s'_2 || s'_3 || s'_4 || s'_5 || s'_6 || a'_1 || a'_2 || y'_1 || y'_2 || m'\}$ . Now,  $V$  and  $V'$  are going to cast another vote by exchanging their parameter(s) from the sets  $(a_1, a_2, y_1, y_2)$  and  $(a'_1, a'_2, y'_1, y'_2)$ , respectively.

1. If  $V'$  borrows  $a_1$  from  $V$ :

So he must also pick up  $s_1, s_2, s_3$  and  $s_4$  from  $V$ 's ticket, since  $H_{lnk} = Hash(a_1 || a_2)$  is involved in the formula of these signed values. Subsequently, he has to borrow  $a_2$  (because of  $H_{lnk}$  and all of the signed values),  $y_1$  (because of  $s_3$ ) and  $y_2$  (because of  $s_4$ ) from  $V$ , too. It can be concluded that  $V'$  can not do anything just replacing all of his parameters with all of the  $V$ 's parameters; in this way, we see that these parameters are the same as the parameters used by the voter  $V$ , so TCS simply detects this forged ticket in the tickets counting phase. Likewise, if he borrows  $a_2$  from  $V$ .

2. If  $V'$  borrows  $y_1$  from  $V$ :

So he has to replace his own  $s'_3$  with  $s_3$ , since  $y_1$  has been participated in the generation of  $s_3$ . By doing so, due to the  $H_{lnk}$ , he must also exchange  $a_1$  and  $a_2$  parameters. Thus, we arrive again at the point that  $V'$  borrows  $a_1$  and leads to nothing. The same argument can be made for  $y_2$ .

Therefore, a voter can not vote again without being detected by exchanging his parameters with another voter. In other words, the proposed attack won't succeed anymore.  $\square$

#### 4. UNRECOGNIZABLE DOUBLE VOTER

Hwang *et al.* have claimed that whenever a double voting is detected in the tickets counting phase, TCS in cooperation with AS can discover the voter who has double voted. In this section we demonstrate that their scheme fails to provide this capability.

Assume that a malicious voter  $V_m$  votes twice with the voting tickets:

$$T = \{s_1 || s_2 || s_3 || s_{4f} || s_5 || s_{6f} || a_1 || a_2 || y_1 || y_{2f} || m\},$$

$$T' = \{s_1 \parallel s_2 \parallel s_3 \parallel s_{4f} \parallel s_5 \parallel s'_{6f} \parallel a_1 \parallel a_2 \parallel y_1 \parallel y_{2f} \parallel m'\}.$$

It means that  $V_m$  applies the same parameters  $(y_1, y_{2f}, a_1, a_2)$  to sign two different voting contents  $m$  and  $m'$ , which is illegal.

In order to create the forged parameters  $(s_{4f}, s_{6f}, y_{2f}, s'_{6f})$ ,  $V_m$  performs the following steps during the voting ticket obtaining phase, while the other parameters are computed in the same way as described in section 2.1:

1.  $V_m$  uses a new random number  $k_f$  in generating the parameter  $w'_2$ :

$$w'_2 = h^{k_f} b_2^{e_{AS}} \bmod n_{AS}.$$

2. The blind signature  $w_7$  is generated by  $AS$  through the equation:

$$\begin{aligned} w_7 &= ((w'_2)^2 \times h^{k_2} \times AS)^{d_{AS}} \bmod n_{AS} \\ &= (h^{2k_f+k_2} \times AS)^{d_{AS}} \times b_2^{-2} \bmod n_{AS} \\ &= (y_{2f} \times AS)^{d_{AS}} \times b_2^{-2} \bmod n_{AS}. \end{aligned}$$

3. Upon receipt of the response from  $AS$ ,  $V_m$  removes the blind factors, specially the one's for  $w_7$  to compute the signature  $s_{4f}$  as the forged  $s_4$ :

$$s_{4f} = w_7 \times b_2^{-2} = (y_{2f} \times AS)^{d_{AS}} \bmod n_{AS}.$$

4.  $V_m$  also calculates the forged  $y_2$  and  $x_2$  as:

$$\begin{aligned} y_{2f} &= h^{2k_f+k_2} \bmod p, \\ x_{2f} &= 2k_f + k_2. \end{aligned}$$

5. Generating the ElGamal digital signatures of two distinct voting contents  $m$  and  $m'$ ,  $V_m$  employs  $x_{2f}$  through the equations:

$$\begin{aligned} s_{6f} &= x_{2f}^{-1}(ma_2 - r) \bmod p-1, \\ s'_{6f} &= x_{2f}^{-1}(m'a_2 - r) \bmod p-1. \end{aligned}$$

Following the above instructions,  $V_m$  constructs the mentioned voting tickets  $T$  and  $T'$ , and sends them to a  $VS$  during the voting and tickets collecting phase. In this phase,  $VS$  successfully validates the forged signatures  $(s_{4f}, s_{6f}, s'_{6f})$  as well as the unchanged ones, and stores the tickets in its database, because:

$$\begin{aligned} AS \times y_{2f} &= s_{4f}^{e_{AS}} \bmod n_{AS}, \\ h^{ma_2} &= y_{2f}^{s_{6f}} \times a_2 \bmod p, \\ h^{m'a_2} &= y_{2f}^{s'_{6f}} \times a_2 \bmod p. \end{aligned}$$

Since the tickets  $T$  and  $T'$  share the same  $(y_1, y_{2f}, a_1, a_2)$  parameters,  $TCS$  can detect the occurrence of double voting in the tickets counting phase, but according to the way that  $V_m$  makes use of the random number  $k_f$  (which is unknown to  $TCS$ ), it is impossible to discover the unique number  $k_2$  as indicated by the following equations:

$$x_1 = \frac{m'a_1 - ma_1}{s'_5 - s_5} \bmod p - 1 = k_1 + k_2,$$

$$x_{2f} = \frac{m'a_2 - ma_2}{s'_{6f} - s_{6f}} \bmod p - 1 = 2k_f + k_2,$$

$$k_1 \neq x_{2f} - x_1,$$

$$k_2 \neq x_1 - k_1.$$

Without the knowledge of  $k_2$ ,  $TCS$  and  $AS$  cannot identify  $V_m$ , so the claimed capability of Hwang *et al.*'s scheme for detection of the malicious double voter fails in this case.

## 5. CONCLUSION

In this paper, we have reviewed the improved e-voting scheme of Hwang *et al.* Then, we have pointed out that Hwang *et al.*'s scheme cannot prevent double/multiple voting of the legitimate registered voters. A clue to solve this weakness has been suggested which employs  $H_{lnk}$  as a link between signed values, and guarantees that these values have to be used all together at the same time. We have further proved that Hwang *et al.*'s scheme is unable to satisfy the claimed capability of identifying the double voters. Scenarios of illustrating the potential double voting as well as unrecognizable double voters have been presented.

Accordingly, Hwang *et al.*'s e-voting scheme and of course the other anonymous e-voting schemes based on Lin *et al.*'s not only cannot resist the double/multiple voting of valid voters, but also disables the ability of Mu and Varadharajan's scheme in discovering malicious double voters.

## REFERENCES

1. Y. Mu and V. Varadharajan, "Anonymous secure e-voting over a network," in *Proceedings of the 14th IEEE Annual Computer Security Applications Conference*, 1998, pp. 293-299.
2. D. Chaum, "Blind signatures system," in *Proceedings of Advances in Cryptology - CRYPTO*, 1983, pp. 153-156.
3. H. Y. Chien, J. K. Jan, and Y. M. Tseng, "Cryptanalysis on Mu-Varadharajan's e-voting schemes," *Applied Mathematics and Computation*, Vol. 139, 2003, pp. 525-530.
4. I. C. Lin, M. S. Hwang, and C. C. Chang, "Security enhancement for anonymous secure e-voting over a network," *Computer Standards and Interfaces*, Vol. 25, 2003, pp. 131-139.
5. C. Yang, C. Lin, and H. Yang, "Improved anonymous secure e-voting over a net-

- work,” *Information Security*, Vol. 15, 2004, pp. 185-191.
6. S. Y. Hwang, H. A. Wen, and T. Hwang, “On the security enhancement for anonymous secure e-voting over computer network,” *Computer Standards and Interfaces*, Vol. 27, 2005, pp. 163-168.
  7. F. Rodriguez-Henriquez, D. Ortiz-Arroyo, and C. Garcia-Zamora, “Yet another improvement over the Mu-Varadharajan e-voting protocol,” *Computer Standards and Interfaces*, Vol. 29, 2007, pp. 471-480.
  8. Y. Y. Chen, J. K. Jan, and C. L. Chen, “The design of a secure anonymous internet voting system,” *Computers and Security*, Vol. 23, 2004, pp. 330-337.
  9. C. A. Neff, “A verifiable secret shuffle and its application to e-voting,” in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, 2001, pp. 116-125.
  10. A. Fujioka, T. Okamoto, and K. Ohta, “A practical secret voting scheme for large scale elections,” in *Proceedings of Advances in Cryptology – AUSCRYPT*, LNCS 718, 1993, pp. 244-251.
  11. D. Chaum, “Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA,” in *Proceedings of Advances in Cryptology – EUROCRYPT*, 1988, pp. 177-182.
  12. T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, Vol. 31, 1985, pp. 469-472.
  13. K. Sampigethaya and R. Poovendran, “A framework and taxonomy for comparison of electronic voting schemes,” *Computers and Security*, Vol. 25, 2006, pp. 137-153.



**Mahdi Asadpour** received his B.S. in Computer Software Engineering from The Sharif University of Technology, Tehran, Iran in 2005. He was a researcher and Linux kernel programmer in Sharif Network Security Center (NSC) from 2002 to 2004, and now is a consultant in Electronic Banking & Security. He has several publications in the areas of network security and multi-agent systems, and has reviewed articles from the Wireless Communication and Mobile Computing (WCMC) and the International Journal of Network Security (IJNS) journals.



**Rasool Jalili** received his Ph.D. in Computer Science from The University of Sydney, Australia in 1995. He then joined the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. He is now an associate professor, doing research in the areas of distributed computing and information security in his network security laboratory; nsc.sharif.edu.