Design and Implementation of a Multiple-Choice E-voting Scheme on Mobile System using Novel *t*-out-of-*n* Oblivious Signature

SHIN-YAN CHIOU AND JIUN-MING CHEN
Department of Electrical Engineering
Chang Gung University
Tao-Yuan, 333 Taiwan
E-mail: {ansel; M9921020}@mail.cgu.edu.tw

Blind signature schemes allow a user to decide any message without disclosing any information about the message to the signer; while oblivious signature schemes allow a signee to selected one of several predetermined messages without revealing any information about the selected message, making such schemes well suited for electronic voting applications. However, the oblivious signature scheme only allows users to select one of the *n* candidates. In this paper, we first propose a *t*-out-of-*n* oblivious signature scheme based on the oblivious transfer method to satisfy the security requirements of not only completeness, unforgeability, privacy, but also selection restriction and non-reduplication; making such scheme well suited for multiple-choice e-voting applications. Moreover, we propose multiple-choice e-voting scheme based on the proposed *t*-out-of-*n* oblivious signature scheme, and implement the scheme in mobile phones to allow users voting securely and conveniently. Security analysis and comparisons of computation and communication efficiency are also provided to validate the proposed schemes.

Keywords: Electronic voting, mobility, blind signature, oblivious signature, security

1. INTRODUCTION

The rapid evolution of network transactions has significantly increased the number of consumers using internet auctions and banking. Protecting user privacy in such applications requires network security technologies, such as network transaction, internet auction, and digital signatures [1-2], which feature several key properties not available with analog signatures, including completeness, unforgeability, undeniability and verifiability. Using public-key cryptography, a signer could sign a message with his exclusive private key. Afterward any verifier can validate the correctness of the signature by using the signer's public key. Thus unlike a traditional signature, a digital signature cannot be forged, not can the signer deny any signature produced by him or her. Digital signatures can thus be treated as authentic validation by the signer, but can be transferred electronically. This technique is very useful in terms of signer authentication, product validation, data integrity assurance and so on.

However, certain situations require protecting the privacy of signature recipients. In 1982, Chaum [3] introduced a blind signature scheme which satisfies this requirement through introducing the property of blindness. In the scheme, a signee could receive a

signed message without revealing any information about the message. Later schemes [4], [5] build on this concept for use in applications including electric payment systems and secure voting systems 6 which require shielding the potentially sensitive content of the requested messages. Mambo et al. [7]–[9] also proposed a new blind signature combined with a proxy signature [10-11] which can be applied to digital voting systems.

Additionally, Chen [12] first proposed the concept of oblivious signatures, a signature scheme that allows a signee to choose one of a set of predetermined messages for signing without the content of the message to the signer. Oblivious signatures and blind signatures both have the property of "signers cannot know the message they sign from signees," but oblivious signatures have one more property, they guarantees the signed message is actually belongs to the predefined set of messages, and any message not belonging to this set will be rejected.

However, Tso et al. [13] pointed out that Chen's proposal [12] did not specifically formalize the scheme or its security properties. As a result they proposed a 1-out-of-n oblivious signature based on Schnorr's scheme [14], and provided formal definitions and security requirements of the oblivious signature scheme including completeness, unforgeability and privacy, making oblivious signatures very well suited for electronic voting applications.

Prior to the development of the oblivious signature concept, Rabin introduced the concept of oblivious transfer [15], a protocol in which the sender sends some subsets of some messages, but does not know which messages the recipient has received. In this way the recipient could obtain the desired message without revealing his preference to the sender, and without the recipient being aware of the other message options. In 2004, Tzeng 16 proposed a 1-out-of-n oblivious transfer and, in 2014 Hao et al. [17] proposed an oblivious transfer scheme based on wireless channel characteristics.

Recently, electronic voting systems are discussed and concerned frequently. More and more countries start to put electronic voting into practice to replace traditional paper voting. For example, in 2005, Estonian announced Estonian E-Voting Laws to adopt electronic voting systems generally. In early days, the digital e-voting systems are usually built based on blind signatures. Some researchers further proposed some signature schemes [6] that combine blind signatures and proxy signatures and applied them to electronic voting systems. However, the application on e-voting systems from blind signatures makes signer unable to know whether the signed message is chosen from one of valid candidates.

Using oblivious signature in e-voting can avoid this kind of problem. Song et al. 18 proposed an electronic voting system based on Tso et al.'s oblivious signature [13]. However, their system increases the loading of voters. Moreover, there is a security problem that attackers can obtain tally result before counting ballots.

In this paper, we first propose a t-out-of-n oblivious signature scheme based on oblivious transfer [16], [17] and satisfy the security requirements of completeness, unforgeability, privacy [13], selection restriction and non-reduplication. Security analysis and comparisons of computation and communication efficiency validate the relative effectiveness of the proposed scheme. The proposed scheme is applied to mobile e-voting system, which is implemented in mobile phones that allow users to vote efficiently, securely and conveniently.

The remainder of this paper is structured as follows: Section 1 provides a brief introduction. Section 2 integrates the relevant literature into a framework for analytical discussion. Section 3 outlines the framework for the proposed *t*-out-of-*n* oblivious signature scheme and provides its security analysis and performance comparison. The security analysis assesses the capability of the proposed framework to ensure completeness, unforgeability, privacy, selection restriction and non-reduplication. This paper compares the effectiveness of the proposed oblivious signature scheme with other similar schemes in terms of computing requirements, transmission loading, and scheme properties. The details for the mobile multiple-choice e-voting application from the proposed *t*-out-of-*n* oblivious signature scheme are presented in Section 4 and our implementation is described in Section 5. Concluding remarks are offered in Section 6.

2. RELATED WORKS

This section introduces the concept and protocols of the proposed signature scheme.

2.1 Oblivious Signature

In 2008, Tso et al. [13] proposed a fair game example to illustrate the operations of the oblivious signature scheme (see Fig. 1). Assume one operator and multiple players. First, one player plays rock, paper scissors (assume he chooses 'rock'). The parameters corresponding to the selected item (i.e., 'rock') are used to calculate *s* which is then transmitted to the operator, but the operator is unable to determine the player's selection based on *s*. After confirming the player's identity, the operator uses *s* to calculate values for rock, paper or scissors, and then signs the calculated values. Based on this determination, the signer transmits his/her signature to the player who can only receive the signature for the item 'rock'. Finally, after the operator announces the solution, the operator determines the winner, and the winner uses the operator's signature to claim his/her prize.

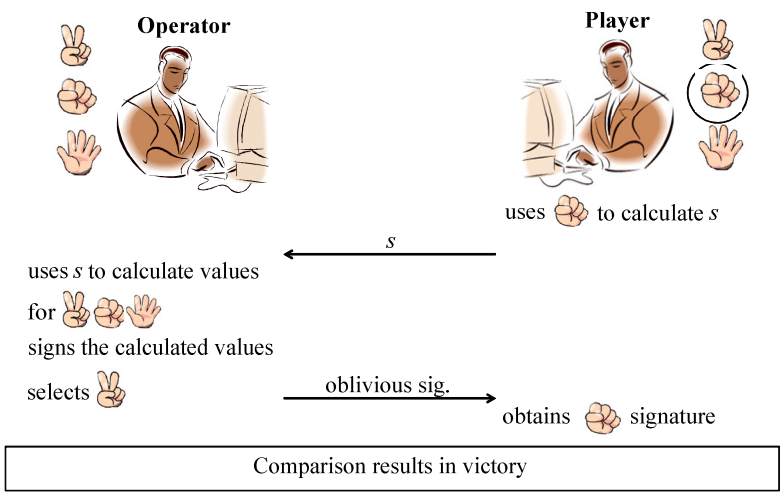


Fig. 1. Schematic illustration of Tso et al.'s oblivious signature

2.2 Chen's Oblivious Signature Agreement

Chen's 1994 oblivious signature concept [12] proposed two types of oblivious signature agreements. The first type uses n keys. In this agreement, the members are n signers (or one signer with n keys) and one or more recipients. The following three characteristics must be met:

(1) By implementing this agreement, the recipient can obtain the signature value of a message, and this signature value uses n to choose a signature key according to the recipient's selection.

(2) Even with possession of this signature key, the signer is unable to determine which is the signing key from the signature value.

(3) In the event of a dispute, others are unable to determine the signature key based on the recipient's signature value.

The second type uses n messages. In this agreement, the members are one signer and one recipient. Assuming both know n messages, the following three conditions must be met:

(1) By implementing this agreement, the recipient can only select a message from n for signing.

(2) The singer is unable to determine which message the recipient has selected.

In the event of a dispute, others are unable to determine which message corresponds to the recipient's signature value.

3. PROPOSED SCHEME OF NOVEL *T*-OUT-OF-*N* OBLIVIOUS SIGNATURE

This section provides a complete introduction to the operational process of the proposed t-out-of-n oblivious signature agreement, and its discussion including security analysis and performance comparison. This agreement allows the recipient to select t messages from n for the signer to sign. If the recipient selects a message which does not belong to n, then the recipient will finally receive a notification from the verifier that the signer cannot be verified. Through this process, the signer is unable to determine which message was selected by the recipient. Table 1 illustrates the notations used in the protocol.

3.1 Attacker model

In our scheme, we assume the channels between the signer and the recipient, the recipient and the verifier, and the signer and the verifier are insecure. Any identity (i.e. the signer, the recipient, or the verifier) communicates with another via an insecure public channel, offering adversaries opportunities to intercept. In the following, we present the assumptions of the attacker model [19, 20].

Table 1 Notation

Symbol	Meaning
(e, N)	RSA public key of signer
(d, N)	RSA private key of signer
Z_N	complete system of residues modulo N
Z_N^*	reduced set of residues modulo N
n	The number of messages
t	The number of chosen messages
m_i	The ith message
a_j	The value of the subscript of the selected message m_i, $j \in (1,t)$
$\sigma(x)$	The signature value of x
$H(\cdot)$	A public cryptographic one- way hash function

(1) An adversary may eavesdrop on all communications between protocol actors over the public channel.

(2) An attacker can modify, delete, resend and reroute the eavesdropped message.

(3) An attacker cannot be a legitimate signer.

(4) The attacker knows the protocol description, which means the protocol is public.

3.2 Security Requirements

This agreement can be applied more flexibly to meet Tso's three security criteria (completeness, unforgeability and privacy) and two more properties (selection restriction and non-reduplication.) The security requirements for the proposed *t*-out-of-*n* oblivious signature scheme are defined as follows.

Definition 1 (Security requirements of the proposed *t*-out-of-*n* oblivious signature scheme) The proposed *t*-out-of-*n* oblivious signature scheme is secure if it achieves (1) Completeness, (2) Unforgeability, (3) Privacy of selected messages, (4) Selection restriction, and (5) Non-reduplication.

The security requirements of our scheme are listed as follows:

(1) *Completeness*: As long as the recipient and the signer can implement the agreement honestly, once the agreement is completed the recipient can obtain the signed message.

(2) *Unforgeability*: Despite the algorithm being publicly published, attackers still have difficulty creating a forged signature within an acceptable time frame.

(3) *Privacy of selected messages*: The signer is unable to determine the recipient's selection.

(4) *Selection restriction*: The recipient is unable to get a valid signature of any message except the *n* messages.

Signer		Recipient
Select $r_i \in_R Z_N^*$, $i = 1,2,...,n$		
Calculate $s_i = H(m_i, r_i)^d \bmod N$	$\xrightarrow{(s_i, r_i, m_i),\ i=1,...,n}$	
		Verify $H(m_i, r_i) \stackrel{?}{=} s_i^e \bmod N$
		If this is correct,
		select $b_j \in_R Z_N^*$, $j = 1,2,...,t$ and $t < n$,
	$\xleftarrow{c_j,\ j=1,...,t}$	calculate $c_j = b_j^e r_{a_j} \bmod N$, $1 \le a_1, a_2, ..., a_t \le n$
Calculate $\beta_j = c_j^d \bmod N$	$\xrightarrow{\beta_j,\ j=1,...,t}$	
		Calculate $v_j = b_j^{-1} \beta_j \bmod N$
		$\sigma(m_{a_j}) = (s_{a_j}, v_j)$

Fig. 2. Signing phase process

(5) *Non-reduplication*: The recipient cannot get more than one signature on the same message in a signing process.

3.3 Protocol of Proposed *t*-out-of-*n* Oblivious Signature Scheme

A complete introduction of the process is provided below, including the roles of the three participants (signer, recipient and verifier) and the three phases (initiation, signing and verification).

(1) *System initiation phase*

This phase first defines parameters and has the signer select an appropriate hash function *H*() and generates the required public key (*e*, *N*) and private key (*d*, *N*) as follows:

Step 1: Select two large prime numbers *p*, *q*

Step 2: Calculate $N = p \times q$

Step 3: Calculate $\phi(N) = (p-1)(q-1)$

Step 4: Select $e \ni GCD(e, \phi(N)) = 1$

Step 5: Calculate $d \ni ed = 1 \bmod \phi(N)$

(2) *Signing phase*

This phase explains how the recipient obtains the signer's complete signature. The process is illustrated in Fig. 2.

Step 1: The signer first randomly selects *i* variables r_i, $i = 1,2,...,n$, and calculates $s_i = H(m_i, r_i)^d \bmod N$, and then transmits $(s_i, r_i, m_i),\ i = 1,2,...,n$ to the recipient.

Step 2: The recipient receives (s_i, r_i, m_i) and then verifies $H(m_i, r_i) \stackrel{?}{=} s_i^e \bmod N$. If it is completely correct, the recipient selects *t* messages from *n* and *t* variables r_{a_j} which correspond to the selected *t* messages, where $j = 1,2,...,t$ and $t < n$, $1 \le a_1, a_2, ..., a_t \le n$. The recipient then selects a random variable b_j, calculates $c_j = b_j^e r_{a_j} \bmod N$ and transmits $c_j,\ j = 1,2,...,t$ to the signer.

Recipient $\xrightarrow{(\sigma(m_{a_j}), m_{a_j})}$ Verifier

Verify $s_{a_j}^e \overset{?}{=} H(m_{a_j}, v_j^e)$

Fig. 3. Verification phase process

Step 3: The signer receives c_j and calculates $\beta_j = c_j^d \bmod N$, before transmitting β_j, $j = 1, 2, ..., t$ to the recipient.

Step 4: The recipient receives β_j, then uses the inverse of b_j to calculate $v_j = b_j^{-1}\beta_j \bmod N$ to obtain the complete signature $\sigma(m_{a_j}) = (s_{a_j}, v_j)$ (where $\sigma(x)$ is the signature value of x).

(3) *Verification phase*

This phase verifies that the signature received by the recipient is correct. The details are illustrated in Fig. 3.

Step 1: The recipient transmits $(\sigma(m_{a_j}), m_{a_j})$ to the verifier.

Step 2: The verifier checks whether $s_{a_j}^e \overset{?}{=} H(m_{a_j}, v_j^e) \bmod N$ is correct. If so, then the signature is verified.

3.4 Security Analysis

This section assesses the security of the proposed method in terms of completeness, unforgeability, privacy, selection restriction and non-reduplication:

(1) *Completeness*

In Step 2 of the signing phase, the recipient verifies $s_{a_j}^e \overset{?}{=} H(m_{a_j}, v_j^e) \bmod N$ to determine the authenticity of the signature, we have $H(m_{a_j}, v_j^e) = H(m_{a_j}, (b_j^{-1}\beta_j)^e) = H(m_{a_j}, (b_j^{-1}c_j^d)^e) = H(m_{a_j}, (b_j^{-1}(b_j^e r_{a_j})^d)^e) = H(m_{a_j}, (b_j^{-1}b_j r_{a_j}^d)^e) = H(m_{a_j}, r_{a_j}) = s_{a_j}^e \bmod N$. Therefore the completeness of the oblivious signature (s_{a_j}, v_j) is proven.

(2) *Unforgeability*

The security of *unforgeability* can be proved via Def. 2, Def. 3, Thm. 1, and Thm. 2.

Definition 2 (RSA problem) Let (e', N') be RSA public keys and $c' = m'^e \bmod N'$, where $m', c' \in Z_N^*$. If m' can be evaluated from given e', N' and c', then we say RSA problem can be solved. (The probability of solving this problem is denoted as $\Pr(m' \mid e', N', c') = \varepsilon_{rsa}$.)

Theorem 1 (Unforgeability) In our protocol, if a recipient can forge the signer's signature, then the RSA problem can be solved.

Proof. A recipient R tries to forge the signer's signature by evaluating β_j from given c_j, e, N. Let RO_1 be a random oracle: input c_j, e and N to output β_j such that

$\beta_j^e = c_j \bmod N$. (i.e. $RO_1(c_j, e, N) \Rightarrow \beta_j : \beta_j^e = c_j \bmod N$). In definition 2, Let $c_j \leftarrow c'$, $e \leftarrow e'$ and $N \leftarrow N'$ be input parameters of RO_1 and obtain output β_j. Let $m' \leftarrow \beta_j$, then m' is evaluated. Therefore, $\Pr(\beta_j \mid c_j, e, N) \le \Pr(m' \mid e', N', c') = \varepsilon_{rsa}$, which means the RSA problem can be solved if RO_1 exists.

Definition 3 (RSA problem under known plaintext attack) Let (e', N') be RSA public keys and $c_i' = m_i'^{e'} \bmod N'$, where $m_i', c_i' \in Z_{N'}^*$ and $i = 1, 2, ..., n+1$. If m_{n+1}' can be evaluated from given $e', N', (m_i', c_i'), c_{n+1}'$, $i = 1, 2, ..., n$, then we say RSA problem under known plaintext attack can be solved. (The probability of solving this problem is denoted as $\Pr(m_{n+1}' \mid e', N', (m_i', c_i'), c_{n+1}') = \varepsilon_{rsakp}$.)

Theorem 2 (Unforgeability under replay attack) In our protocol, if a recipient can forge β_{n+1} from given c_{n+1} and n pairs of (c_i, β_i), $i = 1, 2, ..., n$, then the RSA problem under known plaintext attack can be solved.

Proof. A recipient R tries to forge the signer's signature by evaluating β_{n+1} from given c_{n+1} and n pair of (c_i, β_i). Let RO_2 be a random oracle: input c_{n+1} and n pair of (c_i, β_i) to output β_{n+1} such that $\beta_i^e = c_i \bmod N$, $i = 1, 2, ..., n+1$. (i.e. $RO_5(c_{n+1}, (c_i, \beta_i)) \Rightarrow \beta_{n+1} : \beta_i^e = c_i \bmod N$.) In definition 3, Let $c_{n+1} \leftarrow c_{n+1}'$, $c_i \leftarrow c_i'$, $\beta_i \leftarrow m_i'$, $e \leftarrow e'$ and $N \leftarrow N'$ be input parameters of RO_2 and obtain output β_{n+1}. Let $m_{n+1}' \leftarrow \beta_{n+1}$, then m_{n+1}' is evaluated. Therefore, $\Pr(\beta_{n+1} \mid e, N, (\beta_i, c_i), c_{n+1}) \le \Pr(m_{n+1}' \mid e', N', (m_i', c_i'), c_{n+1}') = \varepsilon_{rsakp}$, which means the RSA problem under known plaintext attack can be solved if RO_2 exists.

(3) *Privacy of selected message*

In Step 2 of the signing phase, the recipient randomly selects a blind factor b_j to blind $c_j = b_j^e r_{a_j} \bmod N$ where c_j is transmitted to the signer. If an attacker intercepts c_j, it is obvious that he/she cannot determine b_j or r_{a_j} from c_j. As for the signer, he/she can decrypt b_j by calculating $b_j = c_j^d r_{a_j}^{-d} \bmod N$, as a result, he/she will obtain n potential b_j corresponding to n random numbers r_{a_j}, since he/she doesn't know which b_j is picked by the recipient, the probability r_{a_j} is selected by the recipient is still $1/n$. Thus the recipient's privacy is perfectly protected.

(4) *Selection restriction*

In Step 2 of the signing phase, the recipient makes his/her selections by choosing t random numbers r_{a_j} and generating c_j. If he/she picks a number r' which is not belong to r_i, $i = 1, 2, ..., n$, he/she will receive $\beta = (b_j^e r')^d \bmod N$ from the signer and the extracted $v_j = b_j^{-1} \beta_j = (r')^d \bmod N$ which will not pass the examining equation

$s_{a_j}^e \overset{?}{=} H(m_{a_j}, v_j^e) \bmod N \quad (\because r' \neq r)$. Moreover, if a recipient attempts to get a signature on an irrelevant message m', since the final signature is composed of $\sigma(m_{a_j}) = (s_{a_j}, v_j)$, where $s_{a_j} = H(m_{a_j}, r_{a_j})^d \bmod N$ is produced by the signer and bound with the message m_{a_j}, the recipient cannot choose a message besides the predetermined messages unless he/she can find the numbers (s', v') that satisfy $(s')^e = H(m', (v')^e) \bmod N$, which contradicts the unforgeability property.

The security of *selection restriction* can be proved via Def. 4, Def. 5, Thm. 3, and Thm. 4.

Definition 4 (Modified RSA signature forgery problem) Let (e, N) be RSA public keys and $s^e = H(m_1, m_2^e) \bmod N$, where $m_1, m_2, s \in Z_N^*$. If (m_2, s) can be evaluated from given (m_1, e, N), then we say modified RSA signature forgery problem can be solved. (The probability of solving this problem is denoted as $\Pr(m_2, s \mid m_1, e, N) = \varepsilon_{mrsasf}$.)

Theorem 3 (Selection restriction) In our protocol, if a recipient can evaluate a signature on an irrelevant message, then the modified RSA signature forgery problem can be solved.

Proof. A recipient R tries to evaluate a signature on an irrelevant message by evaluating (s', v') from given m'. Let RO_3 be a random oracle: input m', e_0 and N_0 to output (s', v') such that $s'^{e_0} = H(m', v'^{e_0}) \bmod N_0$. (i.e. $RO_3(m', e_0, N_0) \Rightarrow (s', v') : s'^{e_0} = H(m', v'^{e_0}) \bmod N_0$.) In definition 4, Let $m' \leftarrow m_1$, $e_0 \leftarrow e$ and $N \leftarrow N$ be input parameters of RO_5 and obtain output (s', v'). Let $s \leftarrow s'$, $m_2 \leftarrow v'$, then (m_2, s) is evaluated. Therefore, $\Pr(s', v' \mid m', e_0, N_0) \leq \Pr(m_2, s \mid m_1, e, N) = \varepsilon_{mrsasf}$, which means the modified RSA signature forgery problem can be solved if RO_3 exists.

Definition 5 (Modified RSA signature forgery problem under known plaintext attack) Let (e, N) be RSA public keys and $s_i^e = H(k_i, m_i^e) \bmod N$, where $k_i, m_i, s_i \in Z_N^*$ and $i = 1, 2, ..., n+1$. If (m_{n+1}, s_{n+1}) can be evaluated from given e, N, k_{n+1} and (k_i, t_i, s_i), where $t_i = m_i^e$, $i = 1, 2, ..., n$, then we say modified RSA signature forgery problem under known plaintext attack can be solved. (The probability of solving this problem is denoted as $\Pr(m_{n+1}, s_{n+1} \mid e, N, k_{n+1}, (k_i, t_i, s_i)) = \varepsilon_{mrsasfkp}$.)

Theorem 4 (Selection restriction under replay attack) In our protocol, if a recipient can evaluate a signature on an irrelevant message from given n triples (m_i, s_i, r_i), where $r_i = v_i^e$, $i = 1, 2, ..., n$, then the modified RSA signature forgery problem under known plaintext attack can be solved.

Proof. A recipient R tries to evaluate a signature by evaluating (s'_{n+1}, v'_{n+1}) from given

m'_{n+1}, e_0, N_0 and n triples (m'_i, s'_i, r'_i). Let RO_4 be a random oracle: input m'_{n+1}, e_0, N_0 and n triples (m'_i, s'_i, r'_i) to output (s'_{n+1}, v'_{n+1}) such that $s'^{e_0}_i = H(m'_i, v'^{e_0}_i) \bmod N_0$, where $r'_i = v'^{e}_i$, $i = 1, 2, ..., n+1$. (i.e. $RO_4(m'_{n+1}, e_0, N_0, (m'_i, s'_i, r'_i)) \Rightarrow (s'_{n+1}, v'_{n+1}) : s'^{e_0}_i = H(m'_i, v'^{e_0}_i) \bmod N_0$.) In definition 5, Let $e_0 \leftarrow e$, $m'_{n+1} \leftarrow k_{n+1}$, $N_0 \leftarrow N$ and $(m'_i, s'_i, r'_i) \leftarrow (k_i, s_i, t_i)$ be input parameters of RO_4 and obtain output (s'_{n+1}, v'_{n+1}). Let $s_{n+1} \leftarrow s'_{n+1}$ and $m_{n+1} \leftarrow v'_{n+1}$, then (m_{n+1}, s_{n+1}) is evaluated. Therefore, $\Pr(s'_{n+1}, v'_{n+1} \mid m'_{n+1}, e_0, N_0, (m'_i, s'_i, r'_i)) \le \Pr(m_{n+1}, s_{n+1} \mid e, N, k_{n+1}, (k_i, t_i, s_i)) = \varepsilon_{mrsasfkp}$, which means the modified RSA signature forgery problem under known plaintext attack can be solved if RO_4 exists.

(5) *Non-reduplication*

If a recipient tries to get two signatures on the same message, he/she can randomly choose b_1, b_2 in the signing phase and calculate $c_1 = b_1^e r_{a_j} \bmod N$, $c_2 = b_2^e r_{a_j} \bmod N$. However, after extracting the signature parameters v_1, v_2, it turns out

$$v_1 = b_1^{-1}\beta_1 = v_2 = b_2^{-1}\beta_2 = r_{a_j}^d \bmod N,$$

the recipient still gets the same signature on the same message, this prevents recipients from getting more than one signature on the same message. Notice that if a normal 1-out-of-n oblivious signature scheme wants to achieve multiple-choice functionality, it will need to repeat the signing phase t times, meanwhile this cannot accomplish the non-reduplication property.

3.5 Performance Comparison

This section provides a performance comparison for the proposed oblivious signature agreement in terms of computation loading, transmission loading and transmission frequency. Such a comparison requires first establishing a reference point in the algorithm. For example, the formulae and algorithm are first run prior to exponentiation, e.g., the algorithm for formula $x^{a \cdot b}$ is first run to calculate $a \cdot b$ before exponentiation, and thus counting for a single exponentiation instance. In addition, "division" can be considered an instance of "exponentiation" because division is a multiplication of inverse, and the inverse calculation uses a Euclidian approach. We then compare the computation loading of the 1-out-of-n oblivious signature scheme, as shown in Table 2, where T_E represents the time required for modular exponentiation.

Table 2 1-out-of-*n* Computation Loading Comparison

Protocol	Chen [12]	Tso [13]	Our Protocol
Recipient (R)	$(2n+10)T_E$	$(2n+2)T_E$	$(n+2)T_E$
Signer (S)	$(3n)T_E$	$(2n+1)T_E$	$(n+1)T_E$
Verifier (V)	$8T_E$	$2T_E$	$2T_E$

Table 3 1-out-of-*n* Transmission Loading

Protocol	Chen [12]	Tso [13]	Our Protocol
$S \rightarrow R$	$n\lvert p\rvert + 3n\lvert q\rvert$	$n\lvert p\rvert + \lvert hash\rvert$	$(2n+1)\lvert N\rvert$
$R \rightarrow S$	$\lvert q\rvert$	$\lvert p\rvert$	$\lvert N\rvert$
$R \rightarrow V$	$4\lvert p\rvert + \lvert q\rvert + \lvert hash\rvert$	$\lvert q\rvert + \lvert hash\rvert$	$2\lvert N\rvert$

Table 4 Comparison of Properties

Protocol	Blind Signature	Chen [12]	Tso [13]	Our Protocol
Blindness	✓	✓	✓	✓
Selection restriction		✓	✓	✓
Multiple choices		(*1)	(*1)	✓
Non-reduplication				✓

As seen in Table 2, the proposed scheme provides the best computation loading of all schemes for all roles (signer, recipient and verifier). Table 3 shows a comparison of the 1-out-of-*n* transmission loading. Because this paper is designed for *t*-out-of-*n* functionality, the proposed protocol is less than ideal. Unlike conventional blind signature schemes, our scheme allows for the selection of multiple choices and provides the properties of non-reduplication, along with lower computation and communication costs (see Table 4). (*1) denotes that the multiple choices provided by Chen [12] or Tso [13] is weak and indirect that it has to proceed the signing phase process twice or more times with no guarantee of twice selection to the same item.

4. DESIGN OF SECURE MULTIPLE-CHOICE E-VOTING SCHEME

In this section, we apply our scheme to a secure mobile e-voting system. The security requirements of the e-voting system are first defined. The system framework and protocol are described. Finally we analyze our protocol according to the defined requirements.

4.1 Attacker model

In our scheme, we assume the channels between the creator and the voter, the voter and the voting center, and the creator and the voting center are insecure. Any identity (i.e. the creator, the voter, or the voting center) communicates with another via an insecure public channel, offering adversaries opportunities to intercept. In the following, we present the assumptions of the attacker model [19, 20].

(1) An adversary may eavesdrop on all communications between protocol actors over

the public channel.

(2) An attacker can modify, delete, resend and reroute the eavesdropped message.

(3) An attacker cannot be a legitimate creator.

(4) The attacker knows the protocol description, which means the protocol is public.

4.2 Security Requirements

The security requirements for the mobile e-voting system are defined as follows.

Definition 6 (Security requirements of the mobile e-voting system) The proposed mobile e-voting system is secure if it achieves (1) Eligibility, (2) Non-reusability, (3) Soundness, (4) Completeness, (5) Verifiability, (6) Fairness, (7) Anonymity, and (8) Non-reduplication.

The security requirements of the mobile e-voting system are as follows:

(1) *Eligibility*. Only eligible voters can cast the votes.
(2) *Non-reusability*. A legitimate voter can vote only once.
(3) *Soundness*. No person can change other persons' vote stealthily.
(4) *Completeness*. All voters can confirm whether their votes are included in total counts.
(5) *Verifiability*. No one can defraud the voting result.
(6) *Fairness*. No one can get any information about the tally result before tally phase.
(7) *Anonymity*. No one can determine any relationship between a vote and a voter.
(8) *Non-reduplication*. No voter can select the same candidate twice.

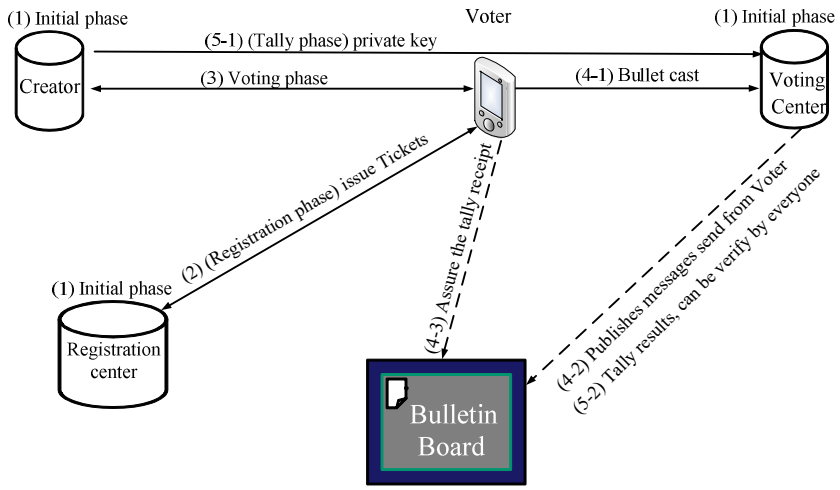


Fig. 4. Voting system

4.3 Introduction of the Proposed Voting System

The scheme (as shown is Fig. 4) involves five entities: Registration center $\mathcal{RC}$, Creator $\mathcal{C}$, Voter $\mathcal{V}$, Voting center $\mathcal{VC}$, and bulletin board $\mathcal{BB}$, where $\mathcal{RC}$ is a trusted party. We assume that the database of Creator exist an ID list of valid voter and there exist a bulletin board which can announce vote information securely. The voting scheme includes five phases: (1) initial phase, (2) registration phase, (3) voting phase, (4) ballot-casting phase, and (5) tally phase.

The detailed steps of the protocol are as follows.

(1) *Initial phase*

In this phase, Registration center (RC) decides a hash function $H(.)$ and each entities (including RC, C, VC) generates their public keys (e_x, N_x) and private keys (d_x, N_x) as follows, where $x =$ RC, C, VC.

Step 1: Choose two large prime numbers p_x, q_x

Step 2: Compute $n_x = p_x \times q_x$

Step 3: Calculate $\phi(N_x) = (p_x - 1)(q_x - 1)$

Step 4: Choose $e_x \in Z^*_{N_x} \ni GCD(e_x, \phi(N_x)) = 1$

Step 5: Compute $d_x \in Z^*_{N_x} \ni e_x d_x = 1 \bmod \phi(N_x)$

(2) *Registration phase*

This phase describes only eligible Voter V_u can obtain valid ticket $Ticket(V_u)$ after qualification checking from the Registration center (RC). The detailed steps of the phase (shown in Fig. 5) are as follows.

Step 1: V_u chooses $pn_u \in_R Z^*_N$ and sends (id_u, pn_u) to RC, where id_u is an identification string of V_u and pn_u is V_u's pseudo-name.

Step 2: After obtaining (id_u, pn_u), RC verifies V_u's identity and voting qualification. If V_u is legitimate, RC computes $sn_u = H(pn_u)^{d_{RC}} \bmod N_{RC}$ and replies $Ticket(V_u) = (sn_u, pn_u)$ to V_u.

Step 3: V_u verify $Ticket(V_u)$ by checking whether the equation $sn_u^{e_{rc}} = H(pn_u) \bmod N_{rc}$ is hold.

Note that V_u only needs to register once for different voting issues.

Registration center $\mathcal{RC}$		Voter $\mathcal{V}_u$
		choose $pn_u \in_R Z^*_N$
$sn_u = H(pn_u)^{d_{RC}} \bmod N_{RC}$	$\xleftarrow{(id_u, pn_u)}$	
$Ticket(\mathcal{V}_u) = (sn_u, pn_u)$	$\xrightarrow{Ticket(\mathcal{V}_u)}$	
		$Ticket(\mathcal{V}_u)$

Fig. 5. Registration phase

Creator $\mathcal{C}$		Voter $\mathcal{V}_u$
		compute
compute $T^{d_C} = Ticket(\mathcal{V}_u) \| tm_u$	$\xleftarrow{ET_u}$	$ET_u = (Ticket(\mathcal{V}_u) \| tm_u)^{e_C} \bmod N_C$
Verify $tm_C - tm_u \overset{?}{\leq} \Delta t$		
and $Ticket(\mathcal{V}_u)$		
choose $r_i \in_R Z_N^*$		
compute $s_i = H(m_i, r_i)^{d_C} \bmod N_C$		
$K_u = h(pn_u)$		
$t_i = E_{K_u}(s_i \| r_i)$		
$(i = 1, 2, ..., n)$	$\xrightarrow{\{(t_i, m_i)\}}$	compute $(s_i' \| r_i') = D_{K_u}(t_i)$
		verify $H(m_i, r_i') \overset{?}{=} s_i'^{e_C} \bmod N_C$
		vote m_{a_j}, $a_j \in \{1, 2, ..., n\}$
		choose $b_j \in_R Z_N^*$,
		compute $c_j = b_j^{e_C} r_{a_j} \bmod N_C$
	$\xleftarrow{\{c_j\}}$	$(i = 1, 2, ..., n; j = 1, 2, ..., t)$
compute $\beta_j = c_j^{d_C} \bmod N_C$	$\xrightarrow{\{\beta_j\}}$	compute $v_j = b_j^{-1} \beta_j \bmod N_C$
		$\sigma(m_{a_j}) = (s_{a_j}, v_j)$

Fig. 6. Voting phase

(3) *Voting phase*

This phase describes the procedures that Voter V$_u$ obtains the signatures $\sigma(m_{a_j})$ of his/her chosen votes m_{a_j}, $j = 1, 2, ..., t$. The detailed steps of the phase (shown in Fig. 6) are as follows.

Step 1: V$_u$ sends $ET_u = (Ticket(\mathcal{V}_u) \| tm_u)^{e_C} \bmod N_C$ to C, where tm_u presents current time.

Step 2: C verifies whether tm_u and $Ticket(\mathcal{V}_u)$ are valid and $Ticket(\mathcal{V}_u)$ is not reused. If it is a valid one without reusing, C stores $Ticket(\mathcal{V}_u)$ in database, chooses $r_i \in_R Z_N^*$, C computes $s_i = H(m_i, r_i)^{d_C} \bmod N_C$, $K_u = h(id_u)$, $t_i = E_{K_u}(s_i \| r_i)$, and sends $\{ (t_i, m_i) \}$ to V$_u$,, i = 1, 2, …, n.

Step 3: V$_u$ computes $(s_i' \| r_i') = D_{K_u}(t_i)$ and verifies $H(m_i, r_i') \overset{?}{=} s_i'^{e_C} \bmod N_C$, i = 1, 2, …, n. If all of them are correct, V$_u$ votes m_{a_j}, chooses $b_j \in_R Z_N^*$,

computes $c_j = b_j^{e_C} r_{a_j} \bmod N_C$, and sends $\{c_j\}$ to C, i = 1, 2, …, n; j = 1, 2, …, t.

Step 4: C computes $\beta_j = c_j^{d_C} \bmod N_C$ and sends $\{\beta_j\}$ to V_u, j = 1, 2, …, t.

Step 5: V_u computes $v_j = b_j^{-1}\beta_j \bmod N_C$ and gets signatures $\sigma(m_{a_j}) = (s_{a_j}, v_j)$, j = 1, 2, …, t.

(4) *Ballot-casting phase*

This phase describes V_u sends his/her votes with signature secretly to the voting center VC. The detailed steps of the phase (shown in Fig. 7) are as follows.

Step 1: V_u computes $B_j^u = (\sigma(m_{a_j}) \| m_{a_j})^{e_C} \bmod N_C$ and $EV_u = (Ticket(\mathcal{V}_u) \square B_j^u)^{e_{VC}} \bmod N_{VC}$, and sends EV_u to VC.

Step 2: VC computes $(Ticket'(\mathcal{V}_u) \| B_j'^u) = D_{d_{VC}}(EV_u)$, verifies $Ticket'(\mathcal{V}_u)$, stores $(Ticket'(\mathcal{V}_u), B_j'^u)$ in database, and publishes $B_j'^u$ on Bulletin board.

Step 3: V_u can check whether his/her votes B_j^u are published on Bulletin board.

(5) *Tally phase*

This phase describes the tally procedures in VC when starting to tally. The detailed steps of the phase (shown in Fig. 8) are as follows.

Step 1: C publishes the private key d_c and sends it VC.

Step 2: VC computes $(\sigma'(m_{a_j}) \| m'_{a_j}) = (B_j'^u)^{d_C} \bmod N_C$, verifies whether the equation $(s'_{a_j})^{e_C} = H(m'_{a_j}, (v'_j)^{e_C}) \bmod N_c$ hold, and publishes d_C, the tally result and B_j^u if which is valid.

Step 3: Each person can verify the validity of B_j^u.

Voter $\mathcal{V}_u$	Voting center $\mathcal{VC}$
$B_j^u = (\sigma(m_{a_j}) \| m_{a_j})^{e_C} \bmod N_C$	
$EV_u = (Ticket(\mathcal{V}_u) \| B_j^u)^{e_{VC}} \bmod N_{VC} \xrightarrow{EV_u}$	compute $(Ticket'(\mathcal{V}_u) \| B_j'^u) = D_{d_{VC}}(EV_u)$
	verify $Ticket'(\mathcal{V}_u)$
	publish $B_j'^u$

Fig. 7. Ballot-casting phase

Creator $\mathcal{C}$	Voting center $\mathcal{VC}$
$\xrightarrow{d_C}$	compute $(\sigma'(m_{a_j}) \| m'_{a_j}) = (B_j'^u)^{d_C} \bmod N_C$
	verify $(s'_{a_j})^{e_C} \stackrel{?}{=} H(m'_{a_j}, (v'_j)^{e_C})$

Fig. 8. Tally phase

4.4 Security Analysis of the Proposed E-Voting

We analyze our protocols according to the requirements defined in Section 4.2.

(1) *Eligibility*. In the registration phase, RC verifies $\mathcal{V}_u$'s voting qualification by his/her identity, only legitimate voters $\mathcal{V}_u$ can get $Ticket(\mathcal{V}_u)$ from RC, and $Ticket(\mathcal{V}_u)$ is required in the voting phase.

(2) *Non-reusability*. In Step 2 of the voting phase, C checks if $Ticket(\mathcal{V}_u)$ is stored in the database, if it is, C aborts the process, preventing $\mathcal{V}_u$ from reusing $Ticket(\mathcal{V}_u)$.

(3) *Soundness*. In the tally phase, all valid ballots are published on the Bulletin board and can be verified publicly with the public key *e*, which prevents invalid ballots.

(4) *Completeness*. In the ballot-casting phase, VC verifies $Ticket(\mathcal{V}_u)$ and publishes B_j^u on Bulletin board if $Ticket(\mathcal{V}_u)$ is legitimate. All $\mathcal{V}_u$ s can check whether his/her ballots are included on Bulletin board to prevent incompleteness.

(5) *Verifiability*. The entire final ballots are posted publicly and can be verified with the public key e_c.

(6) *Fairness*. There is not any person who can obtain any knowledge about the tally before the tally phase. Attackers, including the signer C, cannot know any information about the chosen votes in the voting phase due to the privacy of oblivious signature, of which the security analysis is provided in section 3.4. Moreover, attackers, cannot know any information about the ballot in the ballot-casting phase due to the encryption of messages, of which the security can be proved via Def. 7 and Thm. 5

Definition 7 (partial RSA problem) Let (e,N) be RSA public keys and $c = m^e \bmod N$, where $m,c \in Z_N^*$ and $m = m_1 \| m_2$. If m_2 can be evaluated from given e, N and c, then we say partial RSA problem can be solved. (The probability of solving this problem is denoted as $\Pr(m_2 \mid e,N,c) = \varepsilon_{prsa}$.)

Theorem 5 (Fairness) In our protocol, if a ballot can be obtained in the ballot-casting phase, then the partial RSA problem can be solved.

Proof. An adversary tries to obtain a ballot by evaluating m_{a_j} from given e_C, B_j^u and N_C. Let RO_5 be a random oracle: input e_C, B_j^u and N_C to output m_{a_j} such that $B_j^u = (\sigma(m_{a_j}) \| m_{a_j})^{e_C} \bmod N_C$ (i.e. $RO_5(e_C, B_j^u, N_C) \Rightarrow m_{a_j} : B_j^u = (\sigma(m_{a_j}) \| m_{a_j})^{e_C} \bmod N_C$). In definition 7, Let $e_C \leftarrow e$, $B_j^u \leftarrow c$ and $N_C \leftarrow N$ be input parameters of RO_5 and obtain output m_{a_j}. Let $m_2 \leftarrow m_{a_j}$, then m_2 is evaluated. Therefore, $\Pr(m_{a_j} \mid e_c, N_C, B_j^u) \le \Pr(m_2 \mid e,N,c) = \varepsilon_{prsa}$, which means the partial RSA problem can be solved if RO_5 exists.

(7) *Anonymity*. No one can determine any relationship between a vote and a voter since each voter uses his/her own one-time pseudo name in the whole process.

(8) *Non-reduplication*. If $\mathcal{V}_u$ attempts to vote the same candidate twice, he/she can

randomly choose b_1, b_2 in the voting phase and calculate $c_1 = b_1^e r_{a_j} \bmod N$, $c_2 = b_2^e r_{a_j} \bmod N$. However, after extracting the signature parameters v_1, v_2, it turns out

$$v_1 = b_1^{-1}\beta_1 = v_2 = b_2^{-1}\beta_2 = r_{a_j}^d \bmod N,$$

the voter still gets the same signature on the same candidate, preventing any voter from selecting the same candidate twice.

4.5 Comparison

There are lots of works for e-voting [21-26]. In the section, we compare some properties of the voting schemes in different methods including (A) Traditional voting, (B) direct authorization [21,22], (C) anonymous identifiers [21], (D) blind signatures [21,23], (E) oblivious signatures [12,13], and (Ours) *t*-out-of-*n* oblivious signatures.

As seen in Table 5, the proposed scheme provides the functions for all properties. Unlike conventional blind signature schemes and oblivious signatures, our scheme allows for multiple selections of multiple choices and provides the properties of non-reduplication. Similar with Table 4, (*1) denotes that the property of multiple choices provided by oblivious signatures is weak and indirect that it has to proceed the signing phase process twice or more times with no guarantee of twice selection to the same item.

5. IMPLEMENTATION ON MOBILE SYSTEM

We implement a simulation prototype based on the proposed mobile e-voting scheme on mobile phones running the Android operating system. We use the SHA-256 and the RSA public key system to implement the hash function and encryption/decryption algorithms. The transmission interface is Wi-Fi.

Table 5 Properties Comparison of Different Models of Voting Systems

Protocols	(A)	(B)	(C)	(D)	(E)	(Ours)
Remote access (only)		✓		✓	✓	✓
Anonymity	✓		✓	✓	✓	✓
Results manipulation prevention	✓		✓	✓	✓	✓
Verification of votes			✓	✓	✓	✓
Signature Blindness	N/A		N/A	✓	✓	✓
Selection restriction	✓	✓	✓		✓	✓
t choices out of *n* candidates	✓	✓	✓		(*1)	✓
Non-reduplication	✓	✓	✓			✓

(A) traditional voting,
(B) direct authorization,
(C) anonymous identifiers,
(D) blind signatures,
(E) oblivious signatures,
(Ours) our proposed scheme.

Fig. 9. Mobile phones and Wi-Fi AP

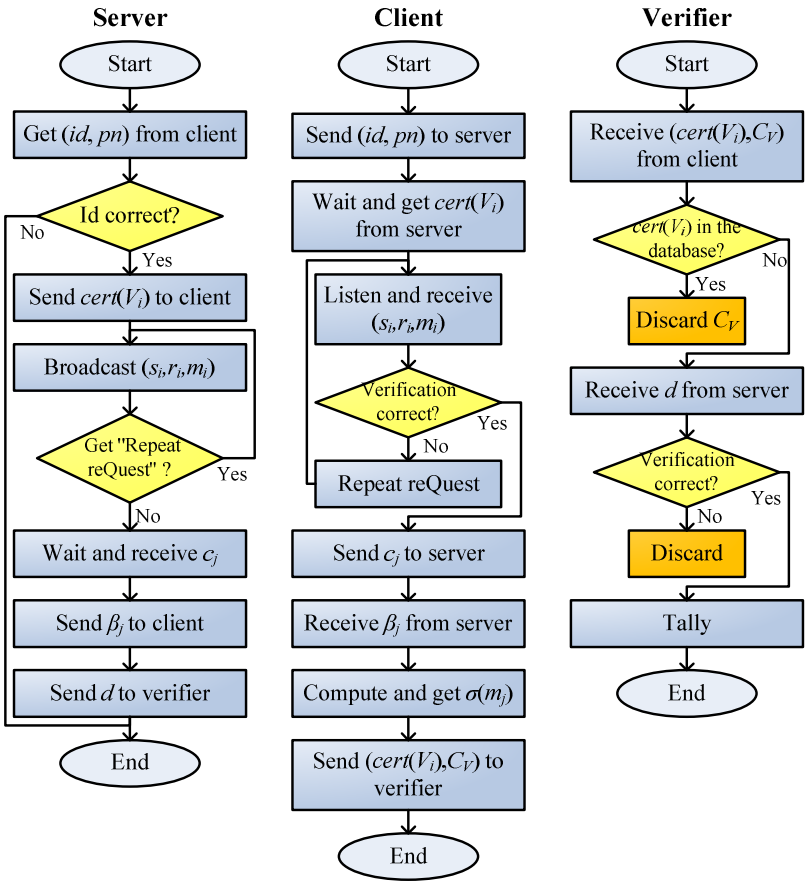
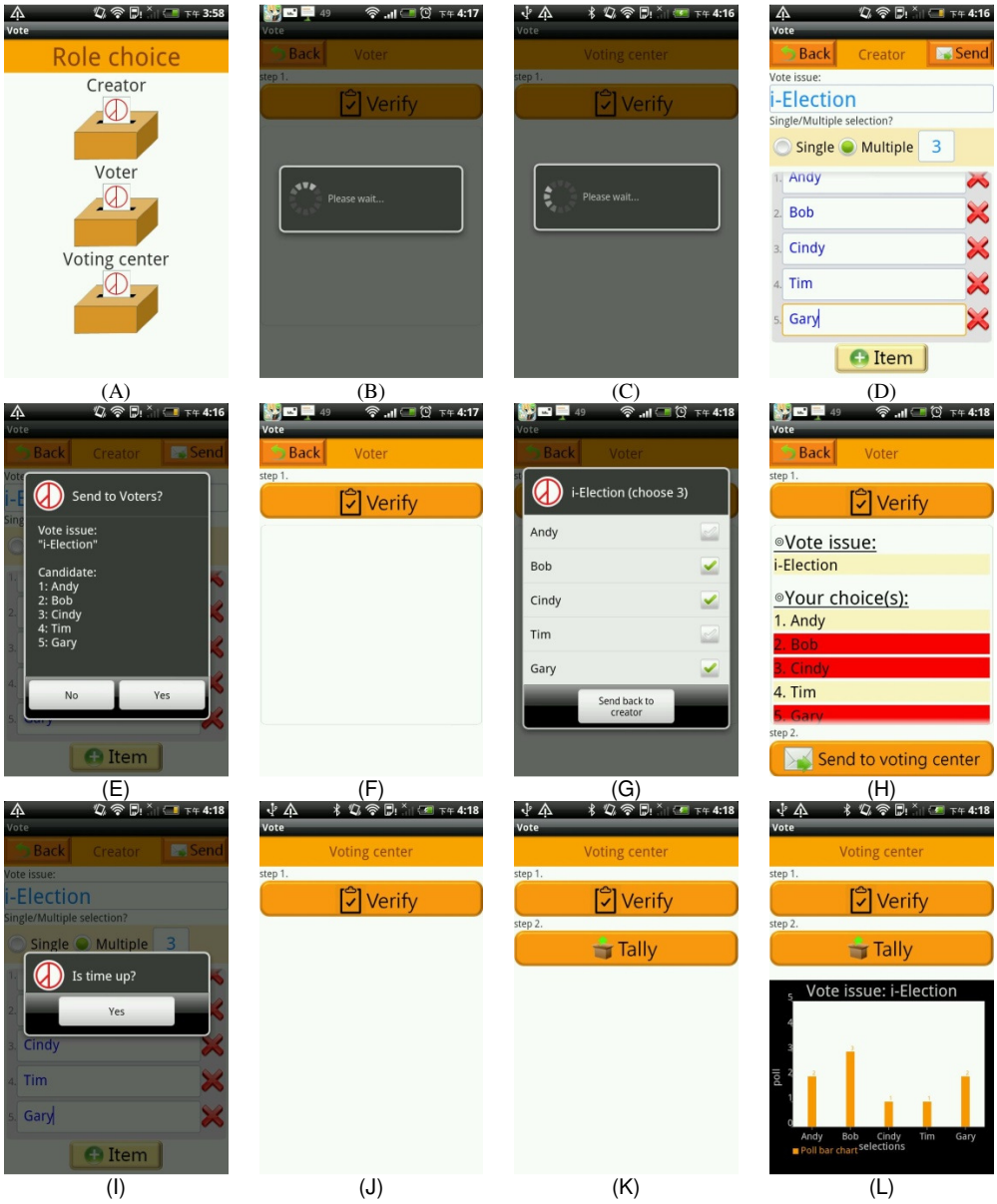


Fig. 10. Application flowchart



(A) (B) (C) (D)

(E) (F) (G) (H)

(I) (J) (K) (L)

(A) Role choice; (B) Voter; (C) Voting center; (D) Creator; (E) Creator sends a vote issue and candidates to voters; (F) Voter receives vote messages from Creator; (G) Voter chooses candidates to vote and sends the message to Creator; (H) Voter obtains signatures; (I) When time is up, Voter presses "Yes" button; (J) Voting center presses "Verify" button to verify the validity of tally; (K) The "Tally" button appears after verification; (L) Poll bar chart of final result.

Fig. 11. Implementation of mobile e-voting system

To implement our proposed scheme, we used a Wi-Fi AP and five cell phones (shown in Fig. 9). Fig. 10 shows the scheme flow in terms of data transmission and Fig.

11 shows the Android mobile phone screens of the JAVA prototype implementation. The hardware included two HTC models (Desire S and Desire HD) both running the Android 2.3.5 Prefessional OS with 1GHz CPU. The implementation included three roles: Creator, Voter, and Voting center.

Fig. 11 (a) illustrates "Role choice" when the application start running. There are three roles, Creator, Voter, and Voting center, can be chosen. If we click "Voter" or "Voting center" (shown in Fig. 11 (b) and Fig. 11 (c)), it presents a listening (or waiting) state to wait the messages sent from Creator. If we want to create a voting, we can choose "Creator", and input Vote issue, single/multiple choice, candidates, etc. Fig. 11 (d) illustrates multiple choice (three choices).

After pressing "Send" and "Yes" (shown in Fig. 11 (e)), Creator sends voting messages to Voter and goes to listening (or waiting) state. After obtaining messages (shown in Fig. 11 (f)), Voter presses "Verify" to verify the validity of the messages, chooses the candidates he/she want to vote (shown in Fig. 11 (g)), and presses "Send back to creator" to send the messages of chosen candidates to Creator for processing oblivious signatures.

After getting messages and processing oblivious signature operations, Creator sends the corresponding massages back to Voter. Next, after Voter gets messages, the chosen candidates are illustrated in red and the button "Send to voting center" appears (shown in Fig. 11 (h)).

Next, Voter press the button "Send to voting center" to wait vote result (under listening mode). When voting time is up (shown in Fig. 11 (i)), Creator can press "Yes" button. After getting the message from Creator (shown in Fig. 11 (j)), Voting center press "Verify" button to verify if the tally is valid. Next, Voting center keeps valid tallies and drops invalid ones, and the button "Tally" appears (shown in Fig. 11 (k)). Finally, voting center press "Tally" button. The poll bar chart of final result appears (shown in Fig. 11 (l)) and this vote finishes.

6. CONCLUSION

This paper constructs a *t*-out-of-*n* oblivious signature scheme that satisfies the security properties of completeness, unforgeability, privacy, selection restriction and non-reduplication. Security analysis and comparisons of computation and communication performance validate the capability and efficiency of the proposed protocol, and its suitability for use in anonymous electronic voting applications. A mobile e-voting protocol using the proposed *t*-out-of-*n* oblivious signature scheme are also proposed and the implementation of the proposed e-voting protocol on Android system mobile devices allows users to securely use the mobile e-voting system conveniently. Future work will focus on decreasing the computation cost for recipients.

REFERENCES

1. H.M. An, S.K. Lee, J.H. Ham, and M.S. Kim, "Traffic identi_cation based on applications using statistical signature free from abnormal tcp behavior," Journal of Information Science and Engineering, Vol. 31, 2015, pp. 1669-1692.

2. Z. Wang and A.D Xia, "Id-based proxy re-signature with aggregate property," Journal of Information Science and Engineering, Vol. 31, 2015, pp. 1199-1211.
3. D. Chaum, “Blind signatures for untraceable payments”, *CRYPTO, Vol. 82, pp.199-203*, 1982.
4. G. Xu, G. Xu, “An ID-based blind signature from bilinear pairing with unlinkability,” *In Consumer Electronics, Communications and Networks (CECNet), 2013 3rd International Conference on (pp. 101-104). IEEE*, 2013.
5. X. Lin, R. Lu, H. Zhu, P. Ho, X. Shen, “Provably Secure Self-certified Partially Blind Signature Scheme from Bilinear Pairings”, *ICC’08, pp. 1530–1535*, 2008.
6. S. Wang, H. Fan, G. Cui, “A proxy blind signature schemes based DLP and applying in e–voting”, *Proceedings of the 7th international conference on Electronic commerce, 2005, pp. 641-645*, 2005
7. M. Mambo, K. Usuda, E. Okamoto, “Proxy signature: Delegation of the power to sign messages”, *IEICE transactions on fundamentals of electronics, communications and computer sciencs, Vol. E79–A, No. 9, pp. 1338–1353*, 1996
8. Z. Dong, H. Zheng, K. Chen, W. Kou, “ID-based proxy blind signature,” *In Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on (Vol. 2, pp. 380-383). IEEE*, 2004.
9. J. Liu, J. Liu, X. Qiu, “A proxy blind signature scheme and an off-line electronic cash scheme,” *Wuhan University Journal of Natural Sciences, 18(2), 117-125*, 2013.
10. W. Hongbin, R. Yan, “A Code-Based Multiple Grade Proxy Signature Scheme,” *In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on (pp. 559-562). IEEE*, 2013.
11. K. S. Kim, D. Hong, I. R. Jeong, “Identity-based proxy signature from lattices,” *Communications and Networks, Journal of, 15(1), 1-7*, 2013.
12. L. Chen, “Oblivious signatures”, *Computer Security–ESORICS’94, Lecture Notes in Computer Science 875, pp.161-172, Springer*, 1994.
13. R. Tso, T. Okamoto, E. Okamoto, “1–out–of–n oblivious signatures”, *Information Security Practice and Experience, pp.45-55, Springer*, 2008.
14. C. P. Schnorr, “Efficient signature generation by smart cards”, *Journal of Cryptology, 4(3), pp.161–174, Springer*, 1991.
15. M. O. Rabin, “How to exchange secrets with oblivious transfer,” *Technical Report, TR-81, Aiken Computation Lab, Harvard University*, 1981.
16. W. G. Tzeng, "Efficient 1-Out-of-*n* Oblivious Transfer Schemes with Universally Usable Parameters," *IEEE Transactions on Computers, vol. 53, no. 2, pp. 232-240, Feb*. 2004.
17. Zhuo Hao; Yunlong Mao; Sheng Zhong; Li, L.E.; Haifan Yao; Nenghai Yu, "Toward Wireless Security without Computational Assumptions—Oblivious Transfer Based on Wireless Channel Characteristics," in *Computers, IEEE Transactions on* , vol.63, no.6, pp.1580-1593, June 2014.
18. C. Song, X. Yin, and Y. Liu, “A practical electronic voting protocol based upon oblivious signature scheme”, *International Conference on Computational Intelligence and Security, pp. 381–384*, 2008.
19. S.Y. Chiou, Z. Ying, J. Liu, “Improvement of a privacy authentication scheme based on cloud for medical environment”, *Journal of Medical Systems*, 40(4), pp.1–15. 2016 ◦

xxxx

20. S.Y. Chiou, "Common friends discovery for multiple parties with friendship ownership and replay-attack resistance in mobile social networks", *Wireless Networks*, pp.1–15. 2016.
21. M. Kucharczyk, "Blind signatures in electronic voting systems", *International Conference on Computer Networks, pp. 349–358*, 2010.
22. J. Epstein, "Electronic Voting", *Computer*, vol.40, no.8, pp.92–95, 2007.
23. S. Ibrahim, M. Kamat, M. Salleh, S.R.A. Aziz, "Secure E-voting with Blind Signature", *NCTT 2003 Proceedings, 4th National Conference on Telecommunication Technology*, Malaysia, 2003.
24. N. Ansari, P.Sakarindr, E. Haghani, C. Zhang, A.K. Jain, Y.Q. Shi, "Evaluating Electronic Voting Systems Equipped with Voter-Verified Paper Records", *IEEE Security & Privacy*, vol.6, no.3, pp.30–39, 2008.
25. D. Chaum, "Secret-Ballot Receipts: True Voter-Verifiable Elections" *IEEE Security & Privacy*, vol.2, no.1, pp.38–47, 2004.
26. N. Paul, A.S. Tanenbaum, "Trustworthy Voting: From Machine to System", *Computer*, vol.42, no.5, pp.23–29, 2009.

Shin-Yan Chiou (邱錫彥) received the PhD degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 2004. From 2004 to 2009, he worked at Industrial Technology Research Institute as a RD Engineer. Since 2009, he joined the faculty of the Department of Electrical Engineering, Chang Gung University, Taoyuan, Taiwan, where he is currently an Associate Professor. He has published a number of journal and conference papers in the areas of information security, social network security and mobile security. His research interests include information security, cryptography, social network security, and secure applications between mobile devices.

Jiun-Ming Chen (陳俊名) received the MS degree in Electrical Engineering from Chang Gung University, Taiwan, in 2013. Since 2013, he joined the faculty of the Shuttle Inc., Taipei, Taiwan, where he is currently an Engineer. His research interests include information security and secure applications between mobile devices.