INFORMATICA, 2007, Vol. 18, No. 4, 535–546

© 2007 *Institute of Mathematics and Informatics, Vilnius*

A Concealed t-out-of-n Signer Ambiguous Signature Scheme with Variety of Keys

Ya-Fen CHANG[1], Chin-Chen CHANG[2,3], Pei-Yu LIN[3]

[1] *Department of Computer Science and Information Engineering*
National Taichung Institute of Technology
Taichung 404, Taiwan, R.O.C.
e-mail: cyf@cs.ccu.edu.tw

[2] *Department of Information Engineering and Computer Science, Feng Chia University*
Taichung 40724, Taiwan, R.O.C.
e-mail: ccc@cs.ccu.edu.tw

[3] *Department of Computer Science and Information Engineering*
National Chung Cheng University
Chiayi 621, Taiwan, R.O.C.
e-mail: lpy91@cs.ccu.edu.tw

Received: July 2005

Abstract. In 2004, Abe *et al.* proposed a threshold signer-ambiguous signature scheme from variety of keys. Their scheme is a generalized case of the ring signature scheme, and it allows the key types to be based on the trapdoor one-way permutations (TOWP) or sigma-protocols including Schnorr's signature scheme. However, the signed message is public for all, which may result in disputes. In this paper, we present a novel threshold signer-ambiguous signature scheme, having the signed message concealed and keeping who the receivers are secret from variety of keys.

Key words: trapdoor one-way permutation, digital signature, signer-ambiguous signature, ring signature, Schnorr's signature scheme.

1. Introduction

Anonymity is an important issue for many applications. As to the digital signature, anonymity may be still demanded even though the digital signature is for authenticating the signer of the corresponding document. In 2001, one motivation for the above scenario comes into being such that one of the possible signers can sign the document without the other possible signers' agreement when the signed document may be harmful if exposed to be public (Rivest *et al.*, 2001). In such schemes, the verifiers know the possible signers instead of the real signer to have the document trustworthy. As a result, the real signer should be ambiguous instead of anonymous. Consequently, it is preferred that the signer-ambiguous signature schemes are setup-free so that the real signer can select the possible signers at will to make himself/herself not be noticed. Contrary to the signer-ambiguous signature schemes, the possible signers are grouped to be a set after the setup process in the threshold signature schemes (Desmedt and Frankel, 1992; Gennaro *et al.*, 1996; Shoup, 2000) and the group signature schemes (Camenisch, 1997; Camenisch and Stadler, 1997; Chaum and Van Heyst, 1991).

Several proposed schemes can be employed as the setup-free signer-ambiguous signature schemes (Gramer *et al.*, 1994; Jakobsson *et al.*, 1996; Rivest *et al.*, 2001). The partial knowledge proof CDS (Gramer *et al.*, 1994) leads the efficient threshold signer-ambiguous schemes, and it can be further combined with other signature schemes based on sigma-protocols – Schnorr's signature scheme for example. Nevertheless, the signature schemes based on TOWP cannot be adopted in CDS such as RSA and Rabin signature schemes (Bellare and Rogaway, 1996; Coron, 2002).

In 2001, Rivest *et al.* proposed the ring signature scheme which almost directly adopts TOWP (Rivest *et al.*, 2001). Later, Bresson *et al.* proposed a t-out-of-n threshold ring signature scheme, where the signature size is exponential to the threshold t (Bresson *et al.*, 2002). Later, a more efficient version was presented such that the signature size is linear to t and n (Kuwakado and Tanaka, 2002). Thereupon, Abe *et al.* presented a modification on the ring signature scheme such that it can be based on both of sigma-protocols and TOWP (Abe *et al.*, 2002). In 2004, Abe *et al.* proposed a t-out-of-n signer-ambiguous signature scheme (Abe *et al.*, 2004). Their scheme allows the key types to be based on the trapdoor one-way permutations (TOWP) or sigma-protocols including Schnorr's signature scheme (Schnorr, 1991). Nevertheless, the signed message is delivered with the signature without being encrypted. With a deep insight into the signer-ambiguous signature schemes, the document may be harmful if exposed to be public. It occurs to us that the document must be concealed. Moreover, for the security of the receivers, no body should know who the receivers are except for the real signers. As a result, we present a novel t-out-of-n signer-ambiguous scheme, which makes the signed message concealed and keeps who the receivers are secret, with variety of keys in this paper.

To illustrate the proposed scheme, let us give a scenario as follows. Suppose that a group of members cooperate to complete one confidential task. However, when one member leaks information to others, it will result in serious damage. At the same time, if some members detect this issue and tend to inform the group manager, what will they do? If these members directly inform the manager by sending a message or a message with the corresponding signature, the manager and each member in the group will know this issue. This approach will place both the informers and the traitor in difficult position. It is because the manager may be the confederate or the traitor may be innocent. Consequently, in this situation, the message must be concealed, and the informers should be anonymous. In our scheme, the informers randomly choose some candidate signers without noticing them and generate the signer-ambiguous signature of the secret message. Since a set of possible signers instead of the real signers are given, the received message is still trustworthy. On the other hand, the secret message is concealed such that the receiver can retrieve it. As a result, the message is still not exposed, and the informers (signers) are still anonymous in our scheme. So it can overcome the problem from which the above scenario suffers. That is, our scheme can prevent unnecessary disputes if the manager may be the confederate or the traitor is innocent.

The rest of this paper is organized as follows. In Section 2, the preliminaries are introduced. In Section 3, Abe *et al.*'s signer-ambiguous signature scheme is introduced. Then our proposed signature scheme is shown in Section 4, followed by some discussions in Section 5. Finally, the conclusions are drawn in Section 6.

2. Preliminaries

In the section, we introduce two types of signature schemes, type-OW and type-3M, which employ TOWP and sigma-protocols, respectively.

2.1. *Type-OW*

Type-OW includes schemes such as the variants of RSA signature scheme, Rabin's signature scheme (Bellare and Rogaway, 1996; Coron, 2002) and Paillier's signature scheme (Paillier, 1999), which use one-way trapdoor permutations. Let a claw-free permutation, F, be a one-way trapdoor permutation and, I, be the corresponding inverse function, where both of F and I are defined over the space C. Let SK and PK be the involved private and public keys, respectively. Suppose that $EM \in C$ is the encoded message. The signature s of EM is $I(SK, EM)$. Note that the verifier may check the equation $EM = F(PK, s)$ to determine if the signature s of EM is valid. What is more, if anyone wants to encrypt EM such that only the owner of the public key needs to compute $Cipher = F(PK, EM)$. Upon getting $Cipher$, the owner computes $EM = I(SK, Cipher)$ to retrieve EM.

2.2. *Type-3M*

Type-3M which is typified by Schnorr's signature scheme includes the signature schemes derived from the sigma protocols. There are three polynomial-time algorithms, A, Z and V, performed by the signer and the verifier. The signer commits to $a \leftarrow A(SK; r)$, which denotes that a is related to r secretly, randomly chooses the challenge c and computes $s = Z(SK, r, c)$. Then the verifier checks whether $a = V(PK, c, s)$ or not to determine the validity of the signature. On the other hand, there are three polynomial-time algorithms, A', E and D, performed by the sender and the receiver. If anyone wants to encrypt the encoded message EM such that only the owner of the public key can get EM, he/she only needs to compute $a' \leftarrow A'(SK; r')$ and $ER = E(PK, r', EM)$. Upon getting ER, the owner computes $EM = D(SK, a', ER)$ to retrieve EM.

3. Abe *et al.*'s Threshold Signer-Ambiguous Signature Scheme

In the following, the details of Abe *et al.*'s scheme are introduced. At first, the initialization is presented. Let the set of the involved public keys be $L = \{PK_1, PK_2, \ldots, PK_n\}$, where the first v public keys in L are of type-OW and the others are of type-3M. Note that at least t corresponding private keys are known to the signers. Let p be a prime larger than any number in the challenge space C_i determined by $PK_i \in L$ for $i = 1, 2, \dots, n$. Let H_0 and H_i be hash functions with the hashing results in Z_p and C_i, respectively, for $i = 1, 2, \dots, n$. The signature scheme consists of two phases, the signature generation phase and the verification phase, described in Subsections 3.1 and 3.2, respectively. In Subsection 3.3, an example is given.

3.1. *The Signature Generation Phase*

Suppose that (L, t, m) are given. The corresponding signature α is generated by the following steps, where m is the signed message.

Step 1: For the real signer U_i, U_i chooses a_i from C_i if U_i's key is of type-OW or commits to $a_i \leftarrow A_i(SK_i; r_i)$ if U_i's key is of type-3M.

Step 2: For other signer U_j who does not sign m, z_j is randomly chosen from Z_p, s_j is chosen from S_j, and c_j and a_j are computed, where S_j is the signature space. If U_j's key is of type-OW, $c_j = H_j(z_j)$ and $a_j = F_j(PK_j, s_j) - c_j$. If U_j's key is of type-3M, $c_j = H_j(z_j)$ and $a_j = V_j(PK_j, c_j s_j)$. Note that the operations in this step are executed by the real signers.

Step 3: $z_0 = H_0(L, t, m, a_1, a_2, \dots, a_n)$ is computed, and an $(n-t)$-degree polynomial P over Z_p is obtained, where $P(i) = z_i$ for the known z_i's.

Step 4: For the real signer U_i, he/she computes $c_i = H_i(P(i))$ and $s_i = I_i(SK_i, a_i + c_i)$ if U_i's key is of type-OW, or he/she computes $c_i = H_i(P(i))$ and $s_i = Z_i(SK_i, r_i, c_i)$ if U_i's key is of type-3M.

Step 5: Finally, the signer-ambiguous signature $\alpha = (P, s_1, s_2, \dots, s_n)$ is obtained.

3.2. *The Verification Phase*

While given (L, t, m) and the signature $\alpha = (P, s_1, s_2, \dots, s_n)$, the verifier does the following steps to verify the signature.

Step 1: If U_i's key is of type-OW, the verifier computes $a_i = F_i(PK_i, s_i) - H_i(P(i)) = F_i(PK_i, s_i) - c_i$.

Step 2: If U_i's key is of type-3M, the verifier computes $a_i = V_i(PK_i, K_i(P(i)), s_i) = V_i(PK_i, c_i, s_i)$.

Step 3: The verifier checks if $P(0) = H_0(L, t, m, a_1, a_2, \dots, a_n)$. If it holds, the verifier confirms that the obtained signature α is valid.

3.3. *An Example of Abe et al.'s Signer-Ambiguous Signature Scheme*

In (Abe *et al.*, 2004), Abe *et al.* presented an example of a t-out-of-n signer-ambiguous signature scheme, where RSA and the Schnorr-like signature schemes are applied, $t = 2$, and $n = 4$. We extend Abe *et al.*'s example such that $t = 2$ and $n = 5$.

Let $G = \{PK_1, PK_2, PK_3, PK_4, PK_5\}$. The key types for U_1, U_2, and U_3 are of RSA signature scheme, and the others are of the Schnorr-like signature scheme. For

$i = 1, 2, 3$, $(SK_i, PK_i) = (d_i, (n_i, e_i))$, where $e_i \in Z_{\phi(n_i)}$ and $d_i = e_i^{-1} \bmod \phi(n_i)$. For $i = 4, 5$, $(SK_i, PK_i) = (x_i, (g_i, q_i, p_i, y_i))$, where g_i is the primitive element with the order q_i and the modulus p_i, q_i is a large prime factor of $\phi(p_i)$, and $y_i = g_i^{x_i} \bmod p_i$. Let p' be a prime greater than n_1, n_2, p_3, p_4 and p_5. Let H_0, H_1, H_2, H_3, H_4 and H_5 be six hash functions with the hashing results in $Z_{p'}$, Z_{n_1}, Z_{n_2}, Z_{n_3}, Z_{q_4}, and Z_{q_5}, respectively.

Suppose that U_1 and U_4 are two real signers who are going to sign the message m. The following procedure is performed.

Step 1: U_1 chooses a_1 from Z_{n_1}. U_4 computes $a_4 = g_4^{r_4} \bmod p_4$.

Step 2: For $i = 2, 3$, z_i is randomly chosen from Z_p, s_i is chosen from Z_{n_i}, and $c_i = H_i(z_i)$ and $a_i = (s_i^{e_i} - c_i) \bmod n_i$ are computed. z_5 is randomly chosen from Z_p, s_5 is chosen from Z_{q_5}, $c_5 = H_5(z_5)$ and $a_5 = g_5^{s_5} y_5^{-c_5} \bmod p_5$ are computed.

Step 3: $z_0 = H_0(L, t, m, a_1, a_2, a_3, a_4, a_n)$ is computed, and a 3-degree polynomial P over $Z_{p'}$ is found, where $P(0) = z_0$, $P(2) = z_2$, $P(3) = z_3$, and $P(5) = z_5$.

Step 4: U_1 computes $c_1 = H_1(P(1))$ and $s_1 = (a_1 + c_1)^{d_1} \bmod n_1$. U_4 computes $c_4 = H_4(P(4))$ and $s_4 = (r_4 + c_4 x_4) \bmod q_4$.

Step 5: Finally, the signer-ambiguous signature $\alpha = (p, s_1, s_2, s_3, s_4, s_5)$ is obtained.

When the verifier wants to verify the signature α, he/she performs as follows:

Step 1: The verifier computes $a_i = (s_i^{e_i} - H_i(P(i))) \bmod n_i$ for $i = 1, 2, 3$.

Step 2: The verifier computes $a_i = g_i^{s_i} y_i^{-K_i(P(i))} \bmod p_i$ for $i = 4, 5$.

Step 3: The verifier checks if $P(0) = H_0(L, t, m, s_1, s_2, s_3, s_4, s_5)$. If it holds, the verifier ensures that the obtained signature α is valid.

4. The Proposed Concealed Threshold Signer-Ambiguous Signature Scheme

In this section, the details of our proposed scheme are presented. Let the set of the involved public keys be $L = \{PK_1, PK_2, \ldots, PK_n\}$, where the first v public keys in L are of type-OW and the others are of type-3M. Note that at least t corresponding private keys are known to the signers. Let p be a prime larger than any number in the challenge space C_i determined by $PK_i \in L$ for $i = 1, 2, \ldots, n$. Let H_0 and H_i be hash functions with the hashing results in Z_p and C_i, respectively for $i = 1, 2, \ldots, n$. For $i = 1, 2, \ldots, n$, P_i denotes the operation field determined by $PK_i \in L$.

The proposed signature scheme consists of two phases, the signature generation phase and the verification-retrieval phase, described in Subsections 3.1 and 3.2, respectively. In Subsection 3.3, an example is given.

4.1. *The Signature Generation Phase*

Suppose that (L, t, M) are given, the corresponding signature α is generated as follows, where M is the message about the signature and it may contain some keywords to be hints for the receivers.

For the keys of type-OW

Step 1: For the real signer U_i, U_i chooses $a_i \leftarrow C_i$.

Step 2: For other signer U_j who is the receiver, the following procedure is executed:
$z'_{2j-1} \leftarrow C_j$;
$z_{2j-1} = z'_{2j-1} + b_{2j-1}P_j$, where $z_{2j-1} \in Z_p$ and $b_{2j-1} \in \{0\} \cup N$;
$z'_{2j} = F_j(PK_j, m_j) - z'_{2j-1}$, where m_j is the message for the receiver U_j;
$z_{2j} = z'_{2j} + b_{2j}P_j$, where $z_{2j} \in Z_p$ and $b_{2j} \in \{0\} \cup N$;
$c_j = H_j(z_{2j-1}||z_{2j})$, where $||$ is the concatenation symbol;
$s_j \leftarrow S_j$, and
$a_j = F_j(PK_j, s_j) - c_j$.
Note that if the receiver U_j's keys are of type-OW, the secret message must be modified irregularly such as appending a random string to it. As a result, m_j's of different receivers with type-OW keys will differ from one another.

Step 3: For other signer U_j who is not the receiver, the following procedure is executed:
$z_{2j-1} \leftarrow Z_p$,
$z_{2j} \leftarrow Z_p$,
$c_j = H_j(z_{2j-1}||z_{2j})$,
$s_j \leftarrow S_j$, and
$a_j = F_j(PK_j, s_j) - c_j$.

For the keys of type-3M

Step 4: For the real signer U_i, U_i chooses $a_i \leftarrow A_i(SK_i; r_i)$.

Step 5: For other signer U_j who is the receiver, the following procedure is executed:
$z'_{2j-1} = a'_j \leftarrow A'_j(SK_j; r_j)$,
$z_{2j-1} = z'_{2j-1} + b_{2j-1}P_j$, where $z_{2j-1} \in Z_p$ and $b_{2j-1} \in \{0\} \cup N$,
$z'_{2j} = E_j(PK_j, r'_j, m_j)$, where m_j is the message for the receiver U_j,
$z_{2j} = z'_{2j} + b_{2j}P_j$, where $z_{2j} \in Z_p$ and $b_{2j} \in \{0\} \cup N$,
$c_j = H_j(z_{2j-1}||z_{2j})$,
$s_j \leftarrow S_j$, and
$a_j = V_j(PK_j, c_j s_j)$.

Step 6: For other signer U_j who is not the receiver, the following procedure is executed:
$z_{2j-1} \leftarrow Z_p$,
$z_{2j} \leftarrow Z_p$,
$c_j = H_j(z_{2j-1}||z_{2j})$,
$s_j \leftarrow S_j$, and
$a_j = V_j(PK_j, c_j s_j)$.
Then the real signer performs as follows.

Step 7: $z_0 = H_0(L, t, M, a_1, a_2, \ldots, a_n)$ is computed, and a $2(n-t)$-degree polynomial P over Z_p is obtained, where $P(i) = z_i$ for the known z_i's.

Step 8: For the real signer U_i, he/she computes $z_{2i-1} = H_i(P(2i-1))$, $z_{2i} = H_i(P(2i))$, $c_i = H_i(z_{2i-1}||z_i)$ and $s_i = I_i(SK_i, a_i+c_i)$ if U_i's key is of type-OW, or he/she computes $z_{2i-1} = H_i(P(2i-1))$, $z_{2i} = H_i(P(2i))$, $c_i = H_i(z_{2i-1}||z_i)$ and $s_i = Z_i(SK_i, r_i, c_i)$ if U_i's key is of type-3M.

Step 9: Finally, the signer-ambiguous signature $\alpha = (P, s_1, s_2, \ldots, s_n)$ is obtained.

4.2. *The Verification-Retrieval Phase*

While given (L, t, M) and the signature $\alpha = (P, s_1, s_2, \ldots, s_n)$, the verifier verifies the signature as follows.

Step 1: If U_i's key is of type-OW, the verifier computes $a_i = F_i(PK_i, s_i) - H_i(P(2i - 1)||P(2i)) = F_i(PK_i, s_i) - c_i$.

Step 2: If U_i's key is of type-3M, the verifier computes $a_i = V_i(PK_i, H_i(P(2i - 1)||P(2i)), s_i) = V_i(PK_i, c_i, s_i)$.

Step 3: The verifier checks if $P(0) = H_0(L, t, M, a_1, a_2, \ldots, a_n)$. If it holds, the verifier is ensured that the obtained signature α is valid.

If the receiver U_j wants to retrieve the encrypted message m_j, he/she executes the following.

For the keys of type-OW

$z'_{2j-1} = P(2j - 1) \bmod P_j$,
$z'_{2j} = P(2j) \bmod P_j$, and
$m_j = I_j(SK_j, z'_{2j} + z'_{2j-1}) \bmod P_j$.

For the keys of type-3M

$z'_{2j-1} = P(2j - 1) \bmod P_j$,
$z'_{2j} = P(2j) \bmod P_j$, and
$m_j = D_j(SK_j, z'_{2j-1}, z'_{2j}) \bmod P_j$.

4.3. *An Example of the Proposed Scheme*

We extend the example in Subsection 3.3, where $t = 2$ and $n = 5$. Let $G = \{PK_1, PK_2, PK_3, PK_4, PK_5\}$. The key types for U_1, U_2, and U_3 are of RSA signature scheme, and the others are of the Schnorr-like signature scheme. For $i = 1, 2, 3$, $(SK_i, PK_i) = (d_i, (n_i, e_i))$, where $e_i \in Z_{\phi(n_i)}$ and $d_i = e_i^{-1} \bmod \phi(n_i)$. For $i = 4, 5$, $(SK_i, PK_i) = (x_i, (g_i, q_i, p_i, y_i))$, where g_i is the primitive element with the order q_i and the modulus p_i, q_i is a great prime factor of $\phi(p_i)$ and $y_i = g_i^{x_i} \bmod p_i$. Let p' be a prime greater than n_1, n_2, p_3, p_4 and p_5. Let H_0, H_1, H_2, H_3, H_4 and H_5 be six hash functions with the hashing results in $Z_{p'}$, Z_{n_1}, Z_{n_2}, Z_{n_3}, Z_{q_4}, and Z_{q_5}, respectively.

Suppose that U_1 and U_4 are two real signers who are going to sign the message M, and U_2 and U_5 are the receivers of m_2 and m_5, respectively. The following steps are performed.

Step 1: U_1 chooses a_1 from Z_{n_1}. U_4 computes $a_4 = g_4^{r_4} \bmod p_4$.

Step 2: For U_2, z'_3 is randomly chosen from Z_{n_2}, and $z_3 = z'_3 + b_3 n_2$ is computed, where $z_3 \in Z_p$ and $b_3 \in \{0\} \cup N$. $z'_4 = (m_2^{e_2} - z'_3) \bmod n_2$ and $z_4 = z'_4 + b_4 n_2$ are computed, where $z_4 \in Z_p$ and $b_4 \in \{0\} \cup N$. $c_2 = H_2(z_3||z_4)$. s_2 is chosen from Z_{n_2}, and $a_2 = (s_2^{e_2} - c_2) \bmod n_2$.

Step 3: For U_3, z_5 and z_6 are randomly chosen from Z_p. $c_3 = H_3(z_5||z_6)$. s_3 is chosen from Z_{n_3}, and $a_3 = (s_3^{e_3} - c_3) \bmod n_3$.

Step 4: For U_5, r'_5 is randomly chosen, where r'_5 is in Z_{q_5} and $z'_9 = a'_5 = g_5^{r'_5} \bmod p_5$. $z_9 = z'_9 + b_9 p_5$ is computed, where $z_9 \in Z_p$ and $b_9 \in \{0\} \cup N$. $z'_{10} = m_5 y_5^{r'_5} \bmod p_5$ and $z_{10} = z'_{10} + b_{10} p_5$ are computed, where $z_{10} \in Z_p$ and $b_{10} \in \{0\} \cup N$. $c_5 = H_5(z_9 || z_{10})$. s_5 is chosen from Z_{q_5}, and $a_5 = g_5^{s_5} y_5^{-c_5} \bmod p_5$.

Step 5: $z_0 = H_0(L, t, M, a_1, a_2, a_3, a_4, a_5)$ is computed, and a 6-degree polynomial P over $Z_{p'}$ is found, where $P(0) = z_0$, $P(3) = z_3$, $P(4) = z_4$, $P(5) = z_5$, $P(6) = z_6$, $P(9) = z_9$ and $P(10) = z_{10}$.

Step 6: U_1 computes $c_1 = H_1(P(1)||P(2))$ and $s_1 = (a_1 + c_1)^{d_1} \bmod n_1$. U_4 computes $c_4 = H_4(P(7)||P(8))$ and $s_4 = (r_4 + c_4 x_4) \bmod q_4$.

Step 7: Finally, the signer-ambiguous signature $\alpha = (P, s_1, s_2, s_3, s_4, s_5)$ is obtained.

When the verifier wants to verify the signature α, he/she performs as follows:

Step 1: The verifier computes $a_i = (s_i^{e_i} - H_i(P(2i-1)||P(2i))) \bmod n_i$ for $i = 1, 2, 3$.

Step 2: The verifier computes $a_i = g_i^{s_i} y_i^{-K_i(P(2i-1)||P(2i))} \bmod p_i$ for $i = 4, 5$.

Step 3: The verifier checks if $P(0) = H_0(L, t, M, s_1, s_2, s_3, s_4, s_5)$. If it holds, the verifier makes sure that the obtained signature α is valid.

As to U_2, he/she retrieves m_2 as follows:
$z'_3 = P(3) \bmod n_2$,
$z'_4 = P(4) \bmod n_2$, and
$m_2 = (z'_3 + z'_4)^{d_2} \bmod n_2$.

As to U_5, he/she retrieves m_5 as follows:
$z'_9 = P(9) \bmod p_5$,
$z'_{10} = P(10) \bmod p_5$, and
$m_5 = ((z'_9)^{x_5})^{-1} z'_{10} \bmod p_5$.

5. Discussions

In this section, we are going to make discussions on our proposed scheme to demonstrate that it is not only secure but also efficient.

Property 1: At least t private keys are known

This property denotes that there are at least t real signers to generate the signature. To determine one k-degree polynomial, $(k+1)$ points are needed. As a result, $(2(n-t)+1)$ points are needed in advance to determine the $2(n-t)$-degree polynomial P. First, the real signers U_i's generate the partial signatures, a_j and s_j, of c_j for U_j's. Second, the real signers U_i's need to choose or compute a_i's in advance. Third, $c_j = H_j(z_{2j-1}||z_{2j})$, $P(2j-1) = z_{2j-1}$ and $P(2j) = z_{2j}$. Forth, $z_0 = H_0(L, t, M, a_1, a_2, \ldots, a_n)$. Since $(2(n-t)+1)$ points are available, the $2(n-t)$-degree polynomial P will be determined at once. Since $c_i = H_i(z_{2i-1}||z_{2i})$ and P have been determined, we have $P(2i-1) = z_{2i-1}$, $P(2i) = z_{2i}$, and $c_i = H_i(z_{2i-1}||z_{2i})$. As a result, the real signers U_i's need to sign c_i's to generate s_i's. Therefore, at least t private keys must be known; otherwise, the valid signer-ambiguous signature cannot be generated.

Property 2: No one can get the knowledge of who the receivers are except for the real signers

In Steps 2 and 5 of the signature generation phase, the secret messages m_j's are all encrypted by the receivers' public keys. And the products z'_{2j-1} and z'_{2j} are all modified to be z'_{2j-1} and z'_{2j}, respectively. Since both of z'_{2j-1} and z'_{2j} are in Z_p, users cannot know who the receivers are. On the other hand, it is wondered whether the receiver can learn the knowledge of other receivers. Suppose that a receiver U_j has obtained m_j. If U_j wants to know whether U_i is the receiver or not, where $i \neq j$, he/she cannot succeed. The reasons are given as follows. If U_i's key is of type-OW, m_i must be different from m_j as shown in Step 2 of the signature generation phase. Moreover, m_i is the product of the secret message modified irregularly, so U_j cannot determine m_i. As a result, even if U_j computes $z'_{2i-1} = P(2i-1) \bmod P_i$, $z'_{2i} = P(2i) \bmod P_i$, and $z''_{2i} = F_i(PK_i, m_j) - z'_{2i-1}$, z''_{2i} and z'_{2i} must be different. If U_i's key is of type-3M, U_j computes $z'_{2i-1} = P(2i-1) \bmod P_i$ and $z'_{2i} = P(2i) \bmod P_i$. Nevertheless, U_j cannot know r'_i to compute $z''_{2i} = E_i(PK_i, r'_i, m_i) \bmod P_i$ and check if z''_{2i} and z'_{2i} are equal to determine whether U_i is the receiver or not.

Property 3: The receiver U_j can retrieve the secret message m_j correctly

In Steps 2 and 5 of the signature generation phase, the secret message m_j has been encrypted by the U_j's public key. Moreover, the polynomial P is determined by all known z''_is including z'_{2j-1} and z'_{2j}. As a result, U_j can get z'_{2j-1} and z'_{2j} to retrieve m_j correctly unless P is modified illegally. Even if P is modified on purpose, U_j can detect easily since the verification of the signature α will fail.

Property 4: The signature size of our proposed scheme is still small

As shown in Section 4, the signature size of our scheme is almost the same as that of Abe *et al.*'s except the polynomial P. The polynomial P is $2(n-t)$-degree in our scheme while it is $(n-t)$-degree in Abe *et al.*'s. That is, the size of the digital signature in our scheme is still proportional to n.

Property 5: Our scheme is efficient

As shown in Sections 3 and 4, the computation load of our scheme is almost the same as that of Abe *et al.*'s. In the signature generation phase, only extra n hash operations, w F function operations, h A' function operations and h E function operations are needed in our scheme, where w is the number of receivers with keys of type-OW and h is the number of receivers with keys of type-3M. It is quite reasonable since the secret message should be encrypted with the receivers' keys, respectively. In the verification-retrieval phase, only extra n hash operations are needed in our scheme.

6. Conclusions

A number of signer-ambiguous signature schemes are proposed to protect the signer, but these schemes cannot keep the essential message secret. This property still results in disputes. In this paper, we have presented a new version which can have the original message concealed and the anonymity of the receivers can also be confirmed at the same time. Moreover, the signature size is still linear to n, and the computation load of our scheme is light as well. In a word, the proposed scheme is secure, efficient, and practical.

References

Abe, M., M. Ohkubo and K. Suzuki (2002). 1-out-of-n signatures from variety of keys. In Y. Zheng (Ed.), *Advances in Cryptology-ASIACRYPT 2002*, Vol. 2501. Springer-Verlag, Germany. pp. 415–432.

Abe, M., M. Ohkubo and K. Suzuki (2004). Efficient threshold signer-ambiguous signatures from variety of keys. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E87-A(2), 471–479.

Bellare, M., and P. Rogaway (1996). The exact security of digital signatures – how to sign with RSA and Rabin. In U. Maurer (Ed.), *Advances in Cryptology – EUROCRYPT'96*, Vol. 1070. Springer-Verlag, Germany. pp. 399–416.

Bresson, E., J. Stern and M. Szydlo (2002). Threshold ring signatures and applications to ad-hoc groups. In M. Yung (Ed.), *Advances in Cryptology – CRYPTO'02*, Vol. 2442. Springer-Verlag, Germany. pp. 465–480.

Camenisch, J. (1997). Efficient and generalized group signatures. In W. Fumy (Ed.), *Advances in Cryptology – EUROCRYPT'97*, Vol. 1233. Springer-Verlag, Germany. pp. 465–479.

Camenisch, J., and M. Stadler (1997). Efficient group signature schemes for large groups. In B. S. Kaliski, Jr. (Ed.), *Advances in Cryptology – CRYPTO'97*, Vol. 1294. Springer- Verlag, Germany. pp. 410–424.

Chaum, D., and E. Van Heyst (1991). Group signatures. In D.W. Davies (Ed.), *Advances in Cryptology – EUROCRYPT'91*, Vol. 547. Springer-Verlag, Germany. pp. 257–265.

Coron, J.S. (2002). Optimal security proofs for PSS and other signature schemes. In L.R. Knudsen (Ed.), *Advances in Cryptology – EUROCRYPT'02*, Vol. 2332. Springer-Verlag, Germany. pp. 272–287.

Desmedt, Y., and Y. Frankel (1992). Shared generation of authenticators and signatures. In J. Feigenbaum (Ed.), *Advances in Cryptology – CRYPTO'91*, Vol. 576. Springer-Verlag, Germany. pp. 457–469.

Gennaro, R., S. Jarecki, H. Krawczyk and T. Rabin (1996). Robust threshold DSS signature. In U. Maurer (Ed.), *Advances in Cryptology – EUROCRYPT'96*, Vol. 1070. Springer-Verlag, Germany. pp. 354–371.

Gramer, R., I. Damgard and B. Schoenmakers (1994). Proofs of partial knowledge and simplified design of witness hiding protocols. In Y.G. Desmedt (Ed.), *Advances in Cryptology – CRYPTO'94*, Vol. 1839. Springer-Verlag, Germany. pp. 174–187.

Jakobsson, M., K. Sako and R. Impagliazzo (1996). Designated verifier proofs and their applications. In U. Maurer (Ed.), *Advances in Cryptology – EUROCRYPT'96*, Vol. 1070. Springer-Verlag, Germany. pp. 143–154.

Kuwakado, H., and H. Tanaka (2002). Digital signature schemes with anonymous signers. *IPSJ SIGNotes Computer Security*, 018-35.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In J. Stern (Ed.), *Advances in Cryptology – EUROCRYPT'99*, Vol. 1592. Springer-Verlag, Germany. pp. 223–238.

Rivest, R., A. Shamir and Y. Tauman (2001). How to leak a secret. In C. Boyd (Ed.), *Advances in Cryptology – ASIACRYPT'01*, Vol. 2248. Springer-Verlag, Germany. pp. 552–565.

Schnorr, C.P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology*, **4**(3), 161–174.

Shoup, V. (2000). Practical threshold signatures. In B. Preneel (Ed.), *Advances in Cryptology – EUROCRYPT'00*, Vol. 1807. Springer-Verlag, Germany. pp. 207–220.

Y.-F. Chang received the BS degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan in 2000. She received her PhD degree in computer science and information engineering in 2005 from National Chung Cheng University, Chiayi, Taiwan. Since 2006, she has been an assistant professor of National Taichung Institute of Technology. Her current research interests include electronic commerce, information security, cryptography, and mobile communications.

C.-C. Chang received his BS degree in applied mathematics in 1977 and his MS degree in computer and decision sciences in 1979, both 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. During the academic years of 1980–1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983–1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From 1989 to 2004, he has worked as a professor in the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. Since 2005, he has worked as a professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taichung, Taiwan. Dr. Chang is a fellow of IEEE, a fellow of IEE 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.

P.-Y. Lin received the MS degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan in 2004. She is currently pursuing her PhD degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan. Her current research interests include digital watermarking, image protection, data mining, and information security.

Paslėpta t-iš-n pasirašančiųjų parašų schema su įvairiais raktais

Ya-Fen CHANG, Chin-Chen CHANG, Pei-Yu LIN

2004 metais Abe ir kiti pasiūlė slenkstinę parašų schemą su įvairiais raktais. Jų schema yra ciklo parašo schemos apibendrinimas ir leidžia naudoti raktus, paremtus vienos krypties kėliniais ar sigma-protokolais, tarp kurių yra ir Schorr'o parašo schema. Tačiau pasirašytas pranešimas yra viešas visiems, dėl ko gali kilti ginčų. Šiame straipsnyje pasiūlyta nauja slenkstinė parašų schema, užtikrinanti pasirašyto pranešimo ir gavėjų slaptumą.