International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6975–6998

A NOVEL DISTRIBUTED RSA BLIND AND WEAKLY UNDENIABLE MULTI-SIGNATURE SCHEME

Chih-Ying Chen[1], Hsiu-Feng Lin[2] and Chiou-Yueh Gun[3]

[1]Department of Communications Engineering
[2]Department of Information Engineering and Computer Science
Feng-Chia University
No. 100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
{ chihchen; hflin }@fcu.edu.tw

[3]Department of Mechanical Engineering
Nan-Kai University of Technology
No. 568, Chung Cheng Rd., Tsaotun Township, Nantou County 54243, Taiwan
moon384@nkut.edu.tw

Received July 2011; revised March 2012

Abstract. *Digital signature is extensively used in many practical applications as numerous cryptographic services have been proposed. In this paper, we will be the first to propose a signature scheme that integrates the following properties. (1) It is a distributed RSA-type multi-signature. (2) It is a blind and weakly undeniable signature. (3) The modulus used by each participant cosigner is identical. (4) Its signature form is identical to that of a standard RSA signature. (5) The length of the signature is independent of the number of participant cosigners. (6) "Zero-knowledge undeniable signatures" is applied to execute the confirmation/disavowal protocols. (7) "The Gap-Problem" is used to solve the forgery problems. A multi-signature distributes the authority and responsibility of the signer among a set of cosigners with efficiency. The blindness and unlinkability properties found in blind signatures can effectively protect the privacy and anonymity of the signature requester. The confirmation and disavowal protocol found in weakly undeniable signatures can safeguard the signature requesters from accepting invalid signatures and reject valid ones. They also protect the signers from false accusations that sign invalid signatures. Therefore, our scheme is suitable for many important cryptographic services, such as electronic voting, electronic payment and other electronic commercial services.*

Keywords: Distributed RSA, Multi-signature, Blind signature, Undeniable signature, Weakly undeniable signature, Integrated digital signature, The Gap-Diffie-Hellman problem

1. **Introduction.** As the methods and structures of network communication activities become increasingly complicated, more and more application systems might require simultaneous support of numerous signing techniques. For example, a sound electronic payment system might need techniques such as blind signatures, undeniable signatures, and multi-signatures at the same time. For the time being, if we use these signing systems individually, various parameters, modulus, signing methods, and verifying procedures each method uses will result in great calculation costs and reduce practical benefits for the entire system. Therefore, an important and urgent research topic in contemporary cryptology has arisen: whether or not it is possible to integrate various signing techniques to use the same parameters and modulus to lower the computational cost and enhance practical benefits of the entire system. Therefore, we present an integrated signing technique mentioned above.

In this paper, we intend to design an integrated signature scheme. We first introduce the concepts of blind signatures, undeniable signatures, weakly undeniable signatures, and multi-signatures, which are all used in our integrated scheme.

The concept of blind signatures was first introduced by Chaum in 1982 [1]. Many solutions with different forms, degrees of security, provability, and properties were suggested in the past years [1-18]. Unlike an ordinary digital signature, a blind signature requires that the signer knows nothing about the contents being signed. This property is called the blindness property [1]. In addition to the blindness property, blind signatures require that even if the signed documents and their corresponding signatures are made public, the signer can only verify the validity of the signature yet cannot trace when and for whom this signature was produced. This property is often referred to as the unlinkability property [1].

Consequently, a blind signature scheme allows efficient protection of the privacy as well as anonymity of a signature requester. Therefore, it plays a central role in many important cryptographic services that emphasize the privacy and anonymity of the users, such as secure electronic voting, electronic check, and electronic cash services, where privacy is of great concern [1,10,19-26].

Besides the blindness and unlinkability properties, recent research on blind signatures emphasizes on developing additional properties such as partial blindness, fair blindness, proxy blindness and low-computation blindness [7,8,10,13-15,18]. The notion of partially blind signature proposed in 1996 allows the signer to explicitly include common information in the blind signature under some agreement with the verifier [7,10,15,27-29].

The undeniable signatures were first introduced by Chaum and Van Antwerpen in 1989 [30]. Many solutions with different forms, degrees of security, provability, and properties were proposed during the past years [30-45]. Unlike the universal verifiability (or self-authenticating) property [32,35] of an ordinary digital signature, the primary feature of undeniable signatures is that a signature can only be verified with the help of the legitimate signer. The verification is achieved through two interaction protocols with the signer – the confirmation protocol and the disavowal (or denial) protocol [30]. The confirmation protocol is used to assure the validity of the signature if it is indeed legitimate. Failing the confirmation test, the disavowal protocol is applied to convince the verifier to reject the validity of the signature. In addition, a cheating signer (even with infinite computing power) has little chance to succeed in validity verifying in both protocols. Therefore, an undeniable signature scheme makes the verification of signatures a valuable operation that protects both the receiver of the signature and the signer. Accordingly, undeniable signatures are more preferable to many commercial or personal sensitive signature applications [25,34,36,46,47]. For example, a software company may attach undeniable signature to its software to insure consumers' possessing legitimate authorization.

It should be pointed out that most previous work on undeniable signature is discrete logarithm based, while only a few are RSA-based (or factoring based) [35,48]. RSA [2] is currently the most widely used algorithm based on the use of PKC, and its security assumption depends on the intractable complexities of factoring (FAC) a large composite integer. The first RSA based scheme was proposed by Gennaro, Krawczyk and Rabin [35]. However, Miyazaki later improved it [49] shortly after it was proposed. The RSA based signature is the most widely accepted type of signature scheme and has the most compact mathematical form. Many recent works on undeniable signature were proposed, trying to achieve properties such as convertibility, delegation, certificateless distribution of signing power and designated confirmer/verifier schemes [31,33,36,38,50-53]. In recent years, the study of convertible undeniable signatures continues to flourish, with its trait introduced by Boyer, Chaum, Damgard and Pederson in 1990, in that the authenticity of confidential data can only be verified with the signer if the data is no longer confidential and can be opened to become publicly verifiable [54-57].

A signature system usually consists of (1) the requester of signature; (2) the signer; (3) the other recipients of the signature. In an ordinary signature system (such as the traditional RSA signature), the signature requester and any recipient of the signature are able to verify the signature by themselves using the signer's public key. Undeniable signatures, however, restrict that any signature recipient, including the original requester, must acquire the approval and cooperation of the original signer in order to verify the signature, hence allowing the signer to control the verifiability of his signatures. Though such a restriction can protect the signer and any recipient of the signature, it appears to be unreasonable to the original requester and increases unnecessary computational cost, unless the requester is the signer himself.

If the requester and the signer are not the same person, a more reasonable and practical restriction is shown: (1) the original requester can verify the signature himself based on a certain mechanism yet cannot prove the validity to any other recipients; (2) any other recipient of the signature can verify the validity only when he gets the approval and cooperation from the original signer. We call a signature technique satisfying the above conditions a "weakly undeniable signature". Apparently, weakly undeniable signatures are more flexible than ordinary undeniable signatures in practice.

In practical applications, many important distributed cryptographic services (e.g., secure electronic payment, electronic voting and electronic bidding) often require several people to cosign a document, either in a broadcasting (or simultaneous) approach or a sequential approach [58], to share the responsibility or distribute the power of generating a signature of the document. A signature generated this way is often called a multi-signature. Multi-signatures cannot be verified without help of all signers, and all signers should be engaged in the verification phase. Over the past years, many solutions and variants have been proposed [3,50,58-65]. The issues mainly concerned include (1) whether the modulus used by each participant signer is identical; (2) whether the size of the multi-signature is independent on the number of signers; (3) whether the signing order needs to be determined in advance.

The remainder of the paper is organized as follows. Section 2 presents an efficient algorithm for generating a set of distributed RSA parameters. Based on these parameters, in Section 3, we propose an integrated signature scheme which is not only a distributed multi-signature, a blind and weakly undeniable signature, but also uses "Zero-knowledge undeniable signatures" to execute the confirmation/disavowal protocols as well. In Section 4, we deal with the properties and discussions of the proposed scheme. The security, which we apply the "The Gap Diffie-Hellman Problem" [66] to solving the security problems, a significant theorem and time complexity are analyzed and discussed in Section 5. Finally, conclusions and future research concerns are given in Section 6.

2. **Generation of a Set of Distributed RSA Parameters.**

Our distributed RSA scheme, like most efficient RSA-based schemes, needs a trusted center to generate a proper set of parameters. In this section, we will briefly introduce an efficient algorithm presented in one of our previous work [67] to generate these parameters. Throughout this paper, all values are integers and the notations used are explained as follows: (a, b) denotes the greatest common divisor of two integers a and b. Accordingly, $(a, b) = 1$ denotes that a is relatively prime to b. Z_n denotes the set of integers between 0 and $n - 1$; Z_n^* denotes the multiplicative group of integer modulo n, i.e., $a \in Z_n$ and $(a, n) = 1$ if $a \in Z_n^*$. $a|b$ denotes a divides b.

Given an integer $r \geq 2$, we can construct two strong primes p and q satisfying $(p-1, q-1) = 2$ and two vectors $< e_1, e_2, \ldots, e_r >$ and $< d_1, d_2, \ldots, d_r >$ satisfying $(d_i, \varphi(n)) = 1$, $(e_i, e_j) \geq A$ if $i \neq j$, $1 \leq i$, $j \leq r$, and $e_1d_1 + e_2d_2 + \cdots + e_rd_r \equiv 1 \pmod{\varphi(n)}$, where $n = p \times q$, $\varphi(n) = (p-1) \times (q-1)$ is the Euler phi-function [68], and A is any prime satisfying $A > 5$ and $A \equiv 1 \pmod 4$. These parameters p, q, n, $\varphi(n)$, e_i and d_i, $1 \leq i \leq r$, will be used in the next section as the system parameters of our proposed signature scheme. (For further details of the following algorithms and theorems, please refer to our pervious work [67,69,70].)

Theorem 2.1. *Dirichlet's Theorem on Primes in Arithmetic Progressions (Theorem 2.9, [68]). Let $(A, B) = 1$. Then the arithmetic progression $A\ell + B$, $\ell = 1, 2, \ldots$, contains infinitely many primes.*

For instance, let $A = 5$ and $B = 4$. It is seen that there are infinitely many primes in the form of $5\ell + 4$, e.g., 19, 29, 59, 79, 89, 109, 139, 149 and 179.

Recall that an integer t is a strong prime [71-73], if (1) t is a prime, (2) there are two large primes t_1 and t_2 such that $t_1 \,|t - 1$ and $t_2 \,|t + 1$, and (3) there are four large primes r_1, s_1, r_2, s_2 such that $r_1 \,|t_1 - 1$, $s_1 \,|t_1 + 1$, $r_2 \,|t_2 - 1$ and $s_2 \,|t_2 + 1$. In this case, r_1, s_1, r_2 and s_2 are often called primes of level-3, t_1 and t_2 are called primes of level-2, and t is called a prime of level-1. Obviously, any prime is a prime of level-3. Referring to [71,72], we can find efficient algorithms to generate strong primes.

Theorem 2.2. *Let t be a prime of level-2. Let A be a prime such that $A > 5$ and $A \equiv 1 \pmod 4$. If $A \not| (t-1)(4t-1)$, then there exists a strong prime p which assumes of the form $A\ell + B$ where $(A, B) = 1$, $A > B > 1$ and $A|(B-1)^2 + 1$. Also, p is of the form $2p' + 1$ where p' is a large prime. (For further detail proof of this theorem, please refer to our previous work [67,69,70].)*

The construction of a prime p satisfying Theorem 2.2 can be described as the following algorithm. The correctness of the algorithm has been implicitly shown in the proof of Theorem 2.2.

Algorithm 2.1.
Input: t is a prime of level-2.
Output: A strong prime p, which assumes $A\ell + B$, where $(A, B) = 1$, $A > B > 1$ and $A|(B-1)^2 + 1$, and $2p' + 1$, where p' is a large prime.
Steps:

```
1. Determine a prime $A$, satisfying $A > 5$, $A \equiv 1 \pmod 4$, and $A \not| (t-1)(4t-1)$.
2. Determine an integer $b$, satisfying $0 < b < A$ and $b^2 \equiv -1 \pmod A$.
3. If $4t - 1 \equiv b + 1 \pmod A$
      then go to Step 4
         else go to Step 5.
4. (1) Choose a large prime $\hat{p}$ of the form$(At)\hat{\ell} + (t-1)$.
   (2) If both $2\hat{p} + 1$ and $4\hat{p} + 3$ are primes
          then output $p = 4\hat{p} + 3$, $A$, $B = b + 1$, and $p' = 2\hat{p} + 1$
             else go to (1) and choose another $\hat{p}$.
5. (1) Compute $R = (4t-1) \bmod A$, and find an integer $a$, satisfying $a(R+1) \equiv b + 2 \pmod A$.
   (2) Choose a large prime $\hat{p}$ of the form $(At)\hat{\ell} + (at - 1)$.
   (3) If both $2\hat{p} + 1$ and $4\hat{p} + 3$ are primes
          then output $p = 4\hat{p} + 3$, $A$, $B = b + 1$, and $p' = 2\hat{p} + 1$
             else go to (2) and choose another $\hat{p}$.
```

Example 2.1. *It is easily to check that $t = 18637$ is a prime of level-2. According to Algorithm 2.1, let us first choose $A = 41$. Obviously, we have $A > 5$, $A \equiv 1 \pmod 4$ and $A \not| (t-1)(4t-1)$. Next, $b = 9$ is chosen, and we have $0 < b < A$ and $b^2 \equiv -1 \pmod A$. Since $4t - 1 \equiv b + 1 \pmod A$ does not hold, we compute, according to (1) of Step 5, $R = (4T - 1) \pmod A$ and solve $a(R + 1) \equiv B + 2 \pmod A$ to obtain $a = 38$. Then, we choose $\hat{\ell} = 8$ and compute $\hat{p} = (At)\hat{\ell} + (at - 1) = 6821141$, $2\hat{p} + 1 = 13642283$ and $4\hat{p} + 3 = 27284567$, and thereafter can make sure that $\hat{p}$, $2\hat{p} + 1$ and $4\hat{p} + 3$ are all primes. Consequently, according to Algorithm 2.1, we have that $p = 4\hat{p} + 3 = 27284567$ is a strong prime assuming both $A\ell + B$ and $2p' + 1$, where $A = 41$, $B = b + 1 = 10$ and $p' = 2\hat{p} + 1 = 13642283$.*

Similarly, consider the case in which $t = 32987, A = 41$ and $b = 9$. We can get that $q = 15833759$ is another strong prime which assumes both the form $A\ell + B$ and $2q' + 1$, where $A = 41$, $b = 10$ and $q' = 7916879$.

Algorithm 2.2.
Input: An integer $r \geq 2$ and two primes p and q of the form $A\ell + B$, where $(A, B) = 1$, $A > B > 1$ and $A|(B - 1)^2 + 1$.
Output: Two vectors $< e_1, e_2, \dots, e_r >$ and $< d_1, d_2, \dots, d_r >$ such that $(d_i, \varphi(n)) = 1$, $(e_i, e_j) \geq A$ if $i \neq j$, $1 \leq i, j \leq r$, and $e_1 d_1 + e_2 d_2 + \cdots + e_r d_r \equiv 1 \pmod{\varphi(n)}$, where $n = p \times q$ and $\varphi(n) = (p - 1) \times (q - 1)$.
Steps:

1. Compute $n = p \times q$ and $\varphi(n) = (p - 1) \times (q - 1)$.
2. Choose $d_i \in Z^*_{\varphi(n)}$, $(1 \leq i \leq r)$.
3. Choose $\alpha_i \in Z_{\varphi(n)}$, $i = 1, 2, \dots, r - 1$, such that $\beta = \alpha_1 d_1 + \alpha_2 d_2 + \cdots + \alpha_{r-1} d_{r-1} < \frac{\varphi(n)+1}{A}$.
4. Compute α_r such that $\alpha_r d_r \equiv \frac{\varphi(n)+1}{A} - \beta \pmod{\varphi(n)}$.
5. Let $e_i = A\alpha_i$, $1 \leq i \leq r$.
6. Return $< e_1, e_2, \dots, e_r >$ and $< d_1, d_2, \dots, d_r >$.

Example 2.2. *Supposing $r = 4$, $A = 41$ and $B = 10$ then we have $(A, B) = 1$, $A > B > 1$ and $A|(B - 1)^2 + 1$. According to Algorithm 2.1, we have obtained in Example 2.1 that $p = 27284567$ and $q = 15833759$ are two strong primes in the form of $A\ell + B$ and $(p - 1, q - 1) = 2$. Inputting p and q into Algorithm 2.2, we will produce two vectors $< e_1, e_2, e_3, e_4 >$ and $< d_1, d_2, d_3, d_4 >$.*

(1) Compute $n = p \times q = 432017258297353$ and $\varphi(n) = (p-1) \times (q-1) = 432017215179028$.
(2) Choose $d_1 = 1077411$, $d_2 = 1176211$, $d_3 = 1533131$, and $d_4 = 1977501$ such that $d_i \in Z^*_{\varphi(n)}$ for $1 \leq i \leq 4$.
(3) Choose $\alpha_1 = 2078502$, $\alpha_2 = 2961154$, and $\alpha_3 = 1743516$ such that $\beta = \alpha_1 d_1 + \alpha_2 d_2 + \cdots + \alpha_{r-1} d_{r-1} < \frac{\varphi(n)+1}{A}$.
(4) Solve the equation $\alpha_4 d_4 = \frac{\varphi(n)+1}{A} - (\alpha_1 d_1 + \alpha_2 d_2 + \alpha_3 d_3) \pmod{\varphi(n)}$ to obtain $\alpha_4 = 395634721918377$.
(5) Compute $e_1 = A\alpha_1 = 85218582$, $e_2 = A\alpha_2 = 121407314$, $e_3 = A\alpha_3 = 71484156$, $e_4 = A\alpha_4 = 236386637029421$.
(6) Return $< e_1, e_2, e_3, e_4 >$ and $< d_1, d_2, d_3, d_4 >$.

Given any integer $r \geq 2$, the parameters p, q, n, $< e_1, e_2, \dots, e_r >$, and $< d_1, d_2, \dots, d_r >$ produced from Algorithm 2.2 will serve as a set of proper parameters of a distributed RSA multi-signature scheme that is going to be proposed in the next section. The property of these parameters can be seen as follows. Suppose the values of p and q are kept secretly, neither of which is known to the attacker. Then it is computationally infeasible for an attacker to factor n into the product of p and q directly without knowing any values of other parameters, because both p and q are strong primes. Also, he is unable to factor n even if he learns part (but not all) of (e_i, d_i), $1 \leq i \leq r$, because $e_1d_1 + e_2d_2 + \cdots + e_rd_r \equiv 1 \pmod{\varphi(n)}$. Since $(p-1, q-1) = 2$, it is also computationally infeasible for an attacker to recover the value of x from $y_i \equiv x^{e_i} \pmod{n}$ by applying Simmons and Norris' attack [74]. Furthermore, we suppose that $(e_i, e_j) = a$ and the attacker knows the values of $y_i \equiv x^{e_i} \pmod{n}$ and $y_j \equiv x^{e_j} \pmod{n}$ for some $i \neq j$. Then he can find r and s such that $re_i + se_j = a$ by the Euclidean algorithm [68] first, and then compute $y = y_i^r \times y_j^s = x^{re_i+se_j} = x^a \pmod{n}$. However, x can still not be recovered from y because $a = (e_i, e_j) \geq A > 5$ and $\gcd(p-1, q-1) = 2$. In addition, since $e_1d_1 + e_2d_2 + \cdots + e_rd_r \equiv 1 \pmod{\varphi(n)}$, an attacker is impossible to deduce the value of any d_i from knowing part or even all of values of e_i's, $1 \leq i \leq r$.

It should be pointed out that a pair of strong primes p and q is called a pair of safe primes [60] if $p = 2p' + 1$ and $q = 2q' + 1$, and p' and q' are both primes. The following theorem can depict characteristics of elements of Z_n^*, where $n = p \times q$, p and q are a pair of safe primes.

Theorem 2.3. [8, Lemma 1] *Let $n = p \times q$, where $p < q$, $p = 2p' + 1$, $q = 2q' + 1$, and p, q, p', q', are all prime numbers. Then,*

(1) The order of elements in Z_n^* is one of the set $\{1, 2, p', q', 2p', 2q', p'q', 2p'q'\}$,

(2) Given an element $x \in Z_n^* \backslash \{-1, 1\}$, such that order $(x) < p'q'$, then either $\gcd(x-1, n)$ or $\gcd(x+1, n)$ is a prime factor of n.

Corollary 2.1. *If $x \in Z_n^* \backslash \{-1, 1\}$ such that $\gcd(x-1, n) = 1$ and $\gcd(x+1, n) = 1$, then order $(x) \geq p'q' = \varphi(n)/4$.*

3. The Proposed Scheme.

In this section, we will first propose a distributed RSA blind and weakly undeniable multi-signature scheme. Then we will discuss some of its properties.

Our scheme is carried out by a Key Generation Center (KGC), a signature requester U, a set of r distributed cosigners P_i's, $1 \leq i \leq r$, and a third verifier V. The KGC is responsible for generating the secure system parameters. The details of the scheme are as follows.

3.1. Key generation phase.

Steps:

1. The KGC generates, by applying Algorithm 2.2, a set of parameters $\{n, e_i, d_i | 1 \leq i \leq r\}$ satisfying $(d_i, \varphi(n)) = 1$, $(e_i, e_j) \geq A$ if $i \neq j$, $1 \leq i, j \leq r$, and $e_1d_1 + e_2d_2 + \cdots + e_rd_r \equiv 1 \pmod{\varphi(n)}$, where $n = p \times q$, $p = 2p' + 1$, $q = 2q' + 1$, p, q, p', q' are all primes, then we have p, q are strong primes numbers, and $\min\{p, q\} > n^{\frac{1}{4}}$, and A is a large prime such that $A > 5$ and $A \equiv 1 \pmod{4}$.
2. The KGC chooses a generator g of the largest multiplicative cyclic subgroup C of Z_n^* with order $\frac{\varphi(n)}{2}$ and computes $G_i \equiv g^{(e_i+1)d_i} \pmod{n}$, $1 \leq i \leq r$, and $G \equiv g^{d_1+d_2+\cdots+d_r} \pmod{n}$. (Note that it has been shown by Gennaro et al. [35] that $\gcd(g \pm 1, n) = p$ or q if the order of $g < \varphi(n)/4$).
3. The KGC determines a full domain hash function $H : \{0, 1\}^* \to C$ [75].
4. (1) KGC broadcasts n, g, H, G_i, $1 \leq i \leq r$ and G as public parameters of the system.

   (2) The KGC distributes n, e_i and d_i to each P_i, $1 \leq i \leq r$.

(3) Each P_i, $1 \le i \le r$, publishes n, e_i as his public key; and keeps d_i secretly as his private key.

3.2. **Signature generation and verification (by the requester) phase.** Suppose that the user U requests the group member P_i, $1 \le i \le r$, to cosign a message $m \in Z_n$ blindly and weakly undeniable. There are two kinds of procedure for signature generating. In the first version, the requester can individually verify each cosigner's partial signature. While in the second version, all cosigners' partial signatures are verified as a whole by the requester. The details are respectively stated as follows.

Version 1 of signature generation and verification by the requester

Steps:

1. (Blinding and Requesting)

(1) U secretly determines two large integers b and c, for b, $c < n^{\frac{1}{8}}$ and computes $a = b \times c + 1$ such that $(G_i)^{a+b} \not\equiv 1 \pmod{n}$, $1 \le i \le r$, and a is an odd integer less than $n^{\frac{1}{4}}$.

(2) For each i, $1 \le i \le r$, U computes $M = H(m)$ and blinds M by computing $B_{1i} \equiv g^{e_i+1} M^a \pmod{n}$ and $B_{2i} \equiv g^{e_i+1} M^{-b} \pmod{n}$, and sends (B_{1i}, B_{2i}) to P_i for requesting a partial signature on M.

2. (Signing)

On receiving (B_{1i}, B_{2i}) from U, each P_i, $1 \le i \le r$, computes $w_{1i} \equiv B_{1i}^{d_i} (\mathrm{mod}\, n)$ and $w_{2i} \equiv B_{2i}^{d_i} (\mathrm{mod}\, n)$ as his partial signature on M. Then he sends (w_{1i}, w_{2i}) back to U and stores the blind message $B_{1i}(g^{e_i+1})^{-1} \bmod n \equiv M^a$ as well as some related information of the protocol in his database.

3. (Verifying each individual partial signature)

(1) Upon receiving the pair (w_{1i}, w_{2i}) from P_i, $1 \le i \le r$, U computes $w_{1i}^b \bmod n$, $w_{2i}^a \bmod n$ and $w_i \equiv w_{1i}^b w_{2i}^a \pmod{n}$.

(2) U checks whether

$$w_i \equiv G_i^{a+b} \pmod{n} \text{ holds or not.} \tag{3.1}$$

(3) U accepts that P_i's partial signature is valid and go to Step 4 if (3.1) holds; else, U goes back to Step 2 and requests P_i to sign (B_{1i}, B_{2i}) again.

4. (Generating multi-signature)

After all partial signatures are verified, computes

$$W_1 \equiv \prod_{i=1}^{r} w_{1i} \pmod{n},$$
$$W_2 \equiv \prod_{i=1}^{r} w_{2i} \pmod{n},$$
$$S_1 \equiv (gG)^{-1} W_1 \pmod{n},$$
$$S_2 \equiv (gG)^{-1} W_2 \pmod{n},$$
$$S \equiv S_1 S_2^c \pmod{n}.$$

Finally selects a secret random integer $h \in Z_n^*$, computes $\lambda = h^{-1}a \pmod{n}$, λ is the authentication factor, and accepts (S, λ) as the multi-signature of M from P_i, $1 \le i \le r$.

Version 2 of signature generation and verification by the requester

Steps:

Step 1 and 2 are exactly the same as Version 1.

3. (Verifying all partial signatures)

(1) After receiving all pairs (w_{1i}, w_{2i}) from P_i's, $1 \leq i \leq r$, U computes W_1, W_2 and W as follows.

$$W_1 \equiv \prod_{i=1}^{r} w_{1i} \ (\text{mod} n),$$
$$W_2 \equiv \prod_{i=1}^{r} w_{2i} \ (\text{mod} n),$$
$$W \equiv W_1^b W_2^a \ (\text{mod} n).$$

(2) U checks whether

$$W \equiv (gG)^{a+b} \ (\text{mod} n) \tag{3.2}$$

(3) U accepts the validity of all cosigners' partial signatures and go to Step 4 if (3.2) holds; otherwise, U goes back to Step 2 and requires all cosigners to sign again.

4. (Generating multi-signature)
U computes

$$S_1 \equiv (gG)^{-1} W_1 \ (\text{mod} n),$$
$$S_2 \equiv (gG)^{-1} W_2 \ (\text{mod} n),$$
$$S \equiv S_1 S_2^c \ (\text{mod} n)$$

Finally selects a secret random integer $h \in Z_n^*$ and computes $\lambda = h^{-1}a \ (\text{mod} n)$, and accepts (S, λ) as the multi-signature of M from P_i, $1 \leq i \leq r$.

Throughout the entire signing procedure, the requester will verify whether all the cosigners p_i used their genuine private key d_i for the signature; therefore he can ensure that

$$S = S_1 S_2^c \ (\text{mod} n) \equiv (gG)^{-1} W_1 (gG)^{-c} W_2^c \equiv (gG)^{-1} \prod_{i=1}^{r} B_{1i}^{d_i} (gG)^{-c} \prod_{i=1}^{r} B_{2i}^{cd_i}$$
$$\equiv g^{-(1+d)} g^{(1+d)} M^{ad} g^{-c(1+d)} g^{c(1+d)} M^{-cbd} \equiv M^{ad} M^{-cbd} \equiv M^{(a-cb)d} \equiv M^d \ (\text{mod} n)$$

is a valid signature.

3.3. **Signature verification (by a third party) phase.** The verification of the signature by any party other than the signature requester is achieved through two zero-knowledge interaction protocols with the cosigners – the confirmation protocol and disavowal protocol. The confirmation protocol is used to convince the signature validity. If the confirmation protocol cannot be satisfied (meaning that the signer denies the fact of signing the given signature), then the second protocol, the disavowal protocol, can be carried out to check if there are deceivers among the cosigners. If the disavowal protocol holds, the verifier is convinced that the signature is invalid and will reject it.

Zero-Knowledge. We use zero-knowledge to execute the following confirmation protocol and disavowal protocol [32]. Zero-knowledge proof systems are indispensable wherever there is necessity to prove the truth of a statement without revealing anything more about it. Zero-knowledge proofs involve two parties: the prover who claims that a statement is true, and the verifier who would like to be convinced that the statement is true. The proof is conducted via an interaction between the parties, at the end of the protocol; the verifier is convinced only when the statement is true. If, however, the prover lies about the statement, the verifier will discover the lie with an overwhelming probability. Informally, an interactive proof system is zero-knowledge if during the interaction the verifier gains no information from prover. In particular, having a transcript of an interaction with prover, verifier is not able to play later the role of the prover for somebody else.

Confirmation protocol. To clarify, we use the following notations to describe the interaction protocols.

$\hat{S}$: the signature presented by V for asking verification;
d_i: the genuine signing key of P_i;
$S \equiv M^{\sum_{i=1}^{r} d_i}$ $(\text{mod} n)$: the legitimate signature;
q_i: the secret integer used by P_i during the protocols;
d_i': the key used by P_i during the protocols;
$S' \equiv M^{\sum_{i=1}^{r} d_i'}$ $(\text{mod} n)$;
$G' \equiv g^{\sum_{i=1}^{r} d_i'}$ $(\text{mod} n)$.

The verifier and each cosigner P_i $(1 \le i \le r)$ agree in advance on the confirmation protocol shown as follows. The cosigner P_i we mention here plays the role of the prover shown above. The requester uses the verifier's public key to encrypt h and sends $E_v(h)$ to the verifier V, such that V is capable of obtaining the blind factor $a = h\lambda$ (mod n). Therefore, the verifier can obtain both the blind message M^a and the signature $\hat{S}^a$ asked for verification from the requester. V establishes the signature's validity using the confirmation protocol shown in Figure 1.

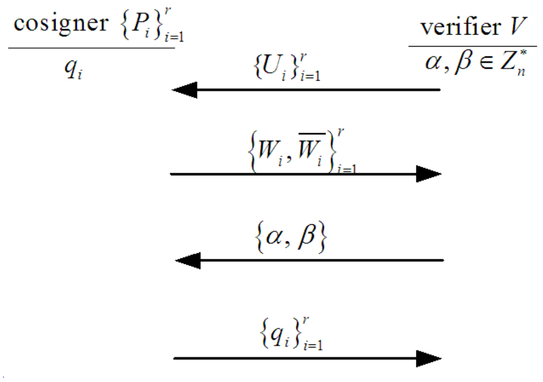
cosigner $\{P_i\}_{i=1}^{r}$
q_i
verifier V
$\alpha, \beta \in Z_n^*$
$\{U_i\}_{i=1}^{r}$
$\{W_i, \overline{W}_i\}_{i=1}^{r}$
$\{\alpha, \beta\}$
$\{q_i\}_{i=1}^{r}$

FIGURE 1. Confirmation protocol

Steps:

1. [V's turn]

(1) Select two secret random odd numbers α and β in Z_n^* with $1 < \alpha$, $\beta < n^{\frac{1}{4}}$. Since a, β are odd integer and are all less than $n^{\frac{1}{4}}$, $\gcd(a\beta, \varphi(n)) = 1$.

(2) Compute $U_i \equiv g^{(e_i+1)\alpha} \cdot M^{a\beta}$ $(\text{mod} n)$, $1 \le i \le r$. (Since α, β are two random integers in Z_n^*, each U_i, $1 \le i \le r$ would be a random number.)

(3) Send $\{U_i\}_{i=1}^{r}$ to P_i as a challenge, $1 \le i \le r$.

2. [P_i's turn, $1 \le i \le r$]

(1) Each cosigner P_i selects a secret integer q_i in Z_n^*, $1 \le i \le r$.

(2) Compute $W_i \equiv U_i \cdot g^{q_i(e_i+1)} \equiv M^{a\beta} \cdot g^{(e_i+1)(\alpha+q_i)}$ $(\text{mod} n)$ and $\overline{W_i} \equiv W_i^{d_i}$ $(\text{mod} n)$, $1 \le i \le r$.

(3) Send $\left\{W_i, \overline{W}_i\right\}_{i=1}^{r}$ back to V as a response.

That is the indistinguishable property, since the verifier cannot distinguish between the individual partial signature $\overline{W}_i$ and the random number W_i.

3. [V's turn]

(1) Send $\{\alpha, \beta\}$ to each P_i, $1 \le i \le r$.

4. $[P_i$'s turn, $1 \le i \le r]$
   (1) Reconstruct U_i, $1 \le i \le r$. (That is, P_i retrieves the blind massage M^a from his database to check whether $g^{(e_i+1)\alpha} M^{a\beta} = U_i \bmod n$ holds or not.)
   (2) Send q_i, $1 \le i \le r$ to V. (If the equation holds, send q_i, $1 \le i \le r$, to V; otherwise, cease to execute the protocol.)
5. $[V$'s turn$]$
   (1) First, reconstruct $\{W_i\}_{i=1}^{r}$.
   (2) Compute $\overline{\overline{W_i}} \equiv \overline{W_i} G_i^{-q_i}$ (mod$n$), $1 \le i \le r$.

$$//\overline{\overline{W_i}} = \overline{W_i} G_i^{-q_i} = W_i^{d_i} G_i^{-q_i} = \left(U_i \cdot g^{q_i(e_i+1)}\right)^{d_i} \left(g^{(e_i+1)d_i}\right)^{-q_i} = U_i^{d_i}//$$

   (3) Check whether $(gG)^{-\alpha} \prod_{i=1}^{r} \overline{\overline{W_i}} \equiv \hat{S}^{a\beta}$ (mod$n$) holds or not (3.3)
   (4) Accept the validity of this signature $\hat{S}$ if (3.3) holds; else, go to disavowal protocol.

$$//\prod_{i=1}^{r} \overline{\overline{W_i}} = \prod_{i=1}^{r} U_i^{d_i} = \prod_{i=1}^{r} \left(g^{(e_i+1)\alpha} \cdot M^{a\beta}\right)^{d_i} = g^{\alpha \cdot \sum_{i=1}^{r} (e_i+1)d_i} \cdot M^{a\beta \sum_{i=1}^{r} d_i}$$
$$= (gG)^{\alpha} (M^d)^{a\beta} \equiv (gG)^{\alpha} S^{a\beta} \text{ (mod}n)//$$

Disavowal protocol. The verifier and each cosigner P_i $(1 \le i \le r)$ agree in advance on the disavowal protocol and the protocol is executed $(\ell + 1)$ rounds. In each round, the verifier randomly chooses a number j and sends a data containing the factor j to the cosigner and requests the cosigner to respond with the actual j. Let both sides execute $(\ell + 1)$ rounds. In the first ℓ rounds, the verifier selects $j = 0$ or $j = 1$, while in the last round (i.e., $(\ell + 1)$-th round) $j \in \{0, 1, 2, \ldots, k\}$, where k is a constant mutually agreed by both sides in advance. Then the probability that the alleged cosigners can successfully cheat the verifier (which is to reject a valid signature) is $\frac{1}{2^\ell} \times \frac{1}{k+1}$. The protocol executed is shown in Figure 2.

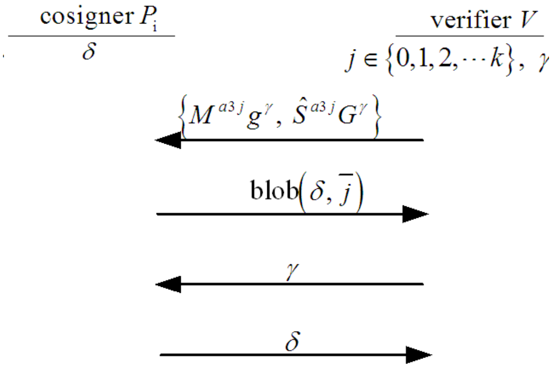
cosigner P_i
δ
verifier V
$j \in \{0,1,2,\cdots k\}$, γ
$\{M^{a3j} g^{\gamma}, \hat{S}^{a3j} G^{\gamma}\}$
blob$(\delta, \overline{j})$
γ
δ

Figure 2. Disavowal protocol

Steps:

1. $[V$'s turn$]$
   (1) Choose an integer j from $\{0, 1, 2, \ldots k\}$, and choose a random number γ in Z_n^* then compute $\left\{M^{a3j} g^{\gamma}, \hat{S}^{a3j} G^{\gamma}\right\}$ for each round.

   There are two phases in this step

   1) In the first phase:
   During the first ℓ rounds, $j \in \{0, 1\}$, and j must be 1 at least once.

2) In the second phase:
On the last round (i.e., $(\ell+1)$-th round), randomly choose

$$j \in \{0, 1, 2, \dots, k\}.$$

(2) Send $\left\{M^{a3j} g^{\gamma}, \hat{S}^{a3j} G^{\gamma}\right\}$ to p_i as a challenge, $1 \le i \le r$.

2. $[p_i$'s turn, $1 \le i \le r]$

(1) All cosigners p_i $(1 \le i \le r)$ mutually decide on a blob $(\delta, \overline{j})$ [4], committing to the value of $\overline{j}$, but hide $\overline{j}$ until the randomly selected δ is revealed.

(2) Send blob $(\delta, \overline{j})$ back to V as a response.

3. $[V$'s turn$]$

(1) Send γ to each p_i, $1 \le i \le r$.

4. $[p_i$'s turn, $1 \le i \le r]$

During the first ℓ rounds, there are at least one time $(M^{a3j} g^{\gamma}) \cdot g^{-\gamma} \equiv M^{a3}$ (since $j = 1$ at least once); on the last round, i.e., the $(\ell+1)$-th round, $(M^{a3j} g^{\gamma}) \cdot g^{-\gamma} \equiv M^{a3j} \pmod{n}$ for some $j \in \{0, 1, 2, \dots, k\}$, send δ to verifier V. Otherwise, p_i's rejects to send δ to V and cease to execute the protocol.

(1) Send δ to verifier V.

5. $[V$'s turn$]$

(1) Open blob $(\delta, \overline{j})$ to get $\overline{j}$;

(2) Check whether $\overline{j} = j$ holds or not; (3.4)

(3) Reject the $\hat{S}$ if (3.4) holds for all $(\ell + 1)$rounds; otherwise, we claim that at least one p_i answered improperly.

4. **Properties and Discussions of the Proposed Scheme.** In this subsection, we will show some properties of our proposed scheme. Also, we will discuss these properties in detail. Our signature system has an interesting integrity property (which means to simultaneously possess the properties of multi-signatures, blind and unlinkable signatures, and weakly undeniable signatures) and various other important properties.

4.1. **Weak undeniability.** In the introduction, we have introduced the properties of weakly undeniable signatures. (1) The requester is able to verify the validity of the signature by himself but cannot prove to any other third party of its validity. (2) Any third party must acquire the approval and cooperation of the original signer in order to verify the validity of the signature. If we wish to prove that the signature is indeed valid, the signer is able to persuade its validity to the verifier via the confirmation protocol; otherwise, the signer uses the disavowal protocol to let the verifier deny its validity. Additionally, no matter which protocol is used and regardless of the computational capability, the probability of a dishonest signer attempting to trick the verifier into accepting an invalid signature or denying a valid one is so small that it can be neglected, and he is usually detected.

The following statements will prove that our system indeed possesses the properties of weakly undeniable signatures.

It should be noted, in the signature generating phase, that our scheme allows the signature requester to verify by himself the legitimacy of all partial signatures and then combine them into a legitimate multi-signature. Yet, he is unable to convince any third party V the validity of the signature, because the values of a, b and c are determined randomly and kept secretly by the requester himself. Even though U informs V the values of a, b and c, V still cannot believe if these values are those U used originally. In fact, U can easily fake a set of $\{a', b', c', M', S'\}$ which can pass the verification of the partial signature as follows. (1) U arbitrarily selects integers a', b', and c' satisfying $a'-1 = b' \times c'$; (2) U computes $w'_{1i} \equiv (G_i)(M')^{a'd'_i}$ (mod$n$), and $w'_{2i} \equiv (G_i)(M')^{-b'd'_i}$ (mod$n$) where d'_i's, $1 \leq i \leq r$, are r integers generated by U; (3) U informs V the values of a', b', and c', and declares that (w'_{1i}, w'_{2i}) is the partial signature of M' and $S' \equiv (M')^{d'_1+d'_2+\cdots+d'_r}$ (mod$n$) is the multi-signature of M' generated by P_i, where $1 \leq i \leq r$. If V accepts the values of a', b', and c', it will convince him that S' is the multi-signature of M' generated by P_i, where $1 \leq i \leq r$. It is due to the fact that (w'_{1i}, w'_{2i}) can pass the verification of the partial signature, because $(w'_{1i})^{b'}(w'_{2i})^{a'} \equiv G_i^{a'+b'}$ (mod$n$). On the other hand, since $S \equiv M^{\sum_{i=1}^{r} d_i}$ (mod$n$) and $e_1d_1 + e_2d_2 + \cdots + e_rd_r \equiv 1$ (mod$\varphi(n)$), it is impossible for any third party to verify the signature S even if he knows all cosigners' public keys $\{e_i | 1 \leq i \leq r\}$. Consequently, he has to conduct some interaction protocols with each of the cosigners in order to make sure the validity or invalidity of the signature. Recall that, this is what we call a "weakly" undeniable signature in introduction.

4.2. **Multi-signature using universal modulus.** According to our signature generating scheme and the signature form $S \equiv M^d$ (mod$n$), $d = \sum\limits_{i=1}^{r} d_i$, our multi-signature scheme has the following important properties. (1) It is an RSA based broadcasting multi-signature system. (2) The form of the signature is identical with general RSA signatures. (3) The length of the signature is independent of the number of signing members. (4) It is worth emphasizing that all members that joined the signing used the same modulus.

The RSA signature is currently the most widely applied signature scheme with the compact mathematical form. However, the trapdoor secrets of general RSA signatures and encryption systems are secretly hidden in the modulus used; therefore, in a multi- or group RSA system, it is an uneasy task to demand each joining member to use the same modulus and ensure the safety of the system simultaneously. Thus, most previously proposed RSA multi-signature systems are sequential multi-signature systems that use different modulus [76,77]. Since each signing member here applies a different modulus, it is inevitable that the computational cost of the system will greatly increase. Due to this fact, considering the benefits in practical application, it would be ideal that all signers used the same modulus. The first main achievement of our paper is that we proposed a highly efficient algorithm (Algorithms 2.1 and 2.2), allowing the key generation center to produce all the necessary security parameters for every signer in a single modulus broadcasting RSA multi-signature system.

4.3. **Blindness and unlinkability.** Obviously, our system also possesses the blindness property of blind signatures (meaning that each signer P_i, $1 \leq i \leq r$, is unaware of the contents when signing a message during the partial signature generation process). This is because during the signature generating process, the requester U first combines the message M and the blind factors a and b into the blind message form $M^a \equiv H(m)^a$ (mod n) and $M^{-b} \equiv H(m)^{-b}$ (mod$n$), then sends (M^a, M^{-b}) to P_i, $1 \leq i \leq r$ in order to sign partial signatures. After the message m and the signature $S \equiv H(m)^d$ (mod$n$) is revealed, $d = \sum\limits_{i=1}^{r} d_i$, it is still impossible to correspond to the blind message M^a. During the signature verification process between cosigners and the verifier, because the protocol applied was zero-knowledge (they have been shown in Conformation protocol and Disavowal protocol), the message $M = H(m)$ and the signature $\hat{S}$ was not leaked to P_i, $1 \leq i \leq r$. Since P_i only knows the signature of the blind message M^a but does not know the parameter a when he cooperates with the verifier to check whether $\hat{S} = S$, P_i

cannot relate M^a with $(m, \hat{S})$, and thus proves that our system possesses the unlinkability property to blind signatures.

4.4. **Confirmability for third party.**

Lemma 4.1. *The protocol of confirmation is zero-knowledge [32].*

Proof: Since α, β are two random integers in Z_n^*, each $U_i = g^{(e_i+1)\alpha} M^{a\beta}$, $1 \le i \le r$ would be a random number. If V sends $\{\alpha, \beta\}$ to each consigner P_i that should result in the message $\{q_i\}_{i=1}^{r}$ being sent, V can form the message $\left\{W_i, \overline{W_i}\right\}_{i=1}^{r}$ determined by any random integer $\{q_i\}_{i=1}^{r}$. Any V not sending such a valid message $\{\alpha, \beta\}$ does not receive the message $\{q_i\}_{i=1}^{r}$, but V can simulate the transcript $\left\{W_i, \overline{W_i}\right\}_{i=1}^{r}$ as $g^t(\mathrm{mod}n)$ and $g^{td_i'}(\mathrm{mod}\, n)$, by choosing t as a random number in Z_n^*, where d_i' is assumed to be the fake secret key of P_i, $1 \le i \le r$. In Step 4, the purpose of reconstructing U_i is to prevent the verifier from packaging a random message $\overline{M}$ with $W_i' \equiv \overline{M}^{a\beta} \cdot g^{(e_i+1)(\alpha+q_i)}(\mathrm{mod}n)$ and using it to trick P_i, allowing the verifier to obtain an extra signature $\overline{W}_i' \equiv (W_i')^{d_i}(\bmod n)$, $1 \le i \le r$. Hence, the protocol is zero-knowledge, namely, on input a message and its valid signature, any verifier V interacting with cosigner P_i, $1 \le i \le r$ learns nothing information aside from the validity of the signature.

Theorem 4.1.
(1) Completeness property

Given that $\hat{S}$ is legitimate (i.e., $\hat{S} = S$), if V and each P_i, $1 \le i \le r$, follow the confirmation protocol, then V always accepts $\hat{S}$ as a valid signature.

(2) Soundness property

It is almost unlikely for cheating cosigners, even computationally unbounded, to convince V to accept an invalid signature.

(3) Non-transferability

The verifier V is unable to convince any other third party V' the validity of the signature without all the cosigners p_i' cooperation.

Proof:
(1) *Completeness property*

Assume that $\hat{S}$ is a valid signature, and each P_i, $1 \le i \le r$ employs his genuine signature key during the proceeding of confirmation protocol. By Equation (3.3)

We have $\hat{S}^{a\beta} \equiv S^{a\beta}(\mathrm{mod}n) \Leftrightarrow (\frac{\hat{S}}{S})^{a\beta} \equiv 1(\mathrm{mod}n)$.

Since $\gcd(a\beta, \varphi(n)) = 1$, $\hat{S} = S$. Hence, accept the validity of $\hat{S}$.

(2) *Soundness property*

Assume that $\hat{S}$ is illegitimate (i.e., $\hat{S} \ne S$), and each key used by P_i during the confirmation protocol is d_i', $1 \le i \le r$. If P_i has the intension to convince V to accept $\hat{S}$ as valid one, they must have $(gG)^{-\alpha} \prod_{i=1}^{r} \overline{\overline{W_i}} \equiv \hat{S}^{a\beta}(\mathrm{mod}n)$, that requires

$$(gG)^{-\alpha}\left(g^{\alpha\sum_{i=1}^{r}(e_i+1)d_i'}\right)\left(M^{a\beta\sum_{i=1}^{r}d_i'}\right) \equiv \hat{S}^{a\beta}(\mathrm{mod}n)$$

or

$$\left(g^{\sum_{i=1}^{r}(e_i+1)(d_i'-d_i)}\right)^{\alpha}\left(\frac{M^{\sum_{i=1}^{r}d_i'}}{\hat{S}}\right)^{a\beta} \equiv 1\ (\mathrm{mod}n) \text{ holds.}$$

Hence, two cases are considered as follows, Case 1: Let $X = \left(g^{\sum\limits_{i=1}^{r}(e_i+1)(d'_i-d_i)}\right)^{\alpha} (\bmod n) \neq 1$, $Y = \left(\frac{M^{\sum\limits_{i=1}^{r}d'_i}}{\hat{S}}\right)^{a\beta} (\bmod n) \neq 1$ be random numbers in Z_n^*, then the probability of $XY \equiv 1(\bmod n)$ will be less then $O(\frac{1}{n})$.

Case 2: If $\sum\limits_{i=1}^{r}(e_i+1)(d'_i-d_i) \equiv 0 \ (\bmod(\varphi))$ and $M^{\sum\limits_{i=1}^{r}d'_i} \equiv \hat{S} \ (\bmod n)$ holds simultaneously, then the probability of $XY \equiv 1 \ (\bmod n)$ will be less than $O(\frac{1}{\varphi(n)}) \cdot O(\frac{1}{n})$

Thus: The probability of V accepting an invalid signature is less then $O(\frac{1}{n})$.

(3) *Non-transferability*

Since V can always simulate and generate a valid transcript. The transcript of our signature generation and verification protocol is shown.

Assume V simultaneously plays the roles of both the verifier and the cosigner, then according to Subsection 3.3, V gives V' a signature $\hat{S} = M^{\sum\limits_{i=1}^{r}d'_i}(\bmod n)$ (The notations d'_i, $1 \leq i \leq r$ and a' are identical to Subsection 4.1) and starts the Confirmation protocol shown in Figure 1:

(i) V and V' both randomly choose two odd integers α' and β' in Z_n^*, and α', $\beta' < n^{\frac{1}{4}}$. Calculate $U'_i \equiv M^{a'\beta'}g^{(e_i+1)\alpha'} \cdot (\bmod n)$ as a challenge and send it to p_i, $1 \leq i \leq r$.

(ii) V impersonates p_i, randomly select integers d'_i, $q'_i \in Z_n^*$, $1 \leq i \leq r$, (d'_i is assumed to be the fake secret key of P_i, q'_i is the fake secret integer randomly selected by P_i) and computes

$$\begin{cases} W'_i \equiv U'_i \cdot g^{q'_i(e_i+1)} \equiv M^{a'\beta'} \cdot g^{(e_i+1)(\alpha'+q'_i)}(\bmod n) \\ \overline{W'_i} \equiv M^{a'\beta'd'} \cdot G_i^{q'_i}G_i^{a'}(\bmod n), \quad 1 \leq i \leq r \end{cases}$$

Send $\left\{W_i, \overline{W_i}\right\}_{i=1}^{r}$ back to V and V' as a response.

That is the indistinguishable property, since the V' cannot distinguish whether $\overline{W'}_i$ is the individual partial signature of W'_i.

(iii) V and V' sends $\{\alpha', \beta'\}$ to p_i, $1 \leq i \leq r$. (Which is actually sent to V).

(iv) V responds with $\{q'_i\}$, $1 \leq i \leq r$.

(v) V and V' verify $W'_i g^{-q'_i(e_i+1)} \equiv U'_i$ and calculates $\overline{\overline{W'_i}} = \overline{W'_i}G_i^{-q'_i} = M^{a'\beta'd'_i} \cdot G_i^{\alpha'}$ to check the verification equation $(gG)^{-\alpha'} \prod\limits_{i=1}^{r} \overline{\overline{W'_i}} \equiv ?\hat{S}^{a'\beta'}(\bmod n)$.

If the verification equations hold, then $\left(M^{\sum\limits_{i=1}^{r}d'_i}\right)^{a'\beta'} \equiv \hat{S}^{a'\beta'}(\bmod n)$, i.e.,

$$\left(\frac{M^{\sum\limits_{i=1}^{r}d'_i}}{\hat{S}}\right)^{a'\beta'} \equiv 1 \ (\bmod n)$$

because $\gcd(a'\beta', \varphi(n)) = 1$, $M^{\sum\limits_{i=1}^{r}d'_i} = \hat{S}$.

4.5. Disavowability for third party.

Lemma 4.2. *The protocol of disavowal is zero-knowledge [32].*

Proof: Any V not supplying an acceptable γ only receives a blob, and so the type of zero-knowledge depends on the type of blob. In Step 4, before the signer p_i sends out δ, the reason he receives γ previously is to check whether M^a sent by the verifier in the disavowal protocol is equal to the blind message M^a in the confirmation protocol. Hence, the protocol is zero-knowledge, namely, on input a message and its valid signature, any verifier V interacting with cosigner p_i, $1 \leq i \leq r$ learns nothing information aside from the fact that $\hat{S}$ is in fact not a valid signature for the message M.

Theorem 4.2.

(1) Completeness property

Assuming that $\hat{S}$ is not valid, if V and each p_i, $1 \leq i \leq r$, follow the disavowal protocol, then V always rejects $\hat{S}$ as a valid signature.

(2) Soundness property

Similarly, it is also impossible for cheating cosigners, even computationally unbounded, to convince V to reject a valid signature.

(3) Non-transferability

The verifier V is unable to convince any other third party V' the invalidity of the signature without the cooperation of all cosigners p_i.

Proof:

(1) *Completeness property*

Assume that $\hat{S} \neq S$, and each p_i, $1 \leq i \leq r$ employs his genuine signature key d_i during the proceeding of disavowal protocol, we have

$$W_i \equiv (M^{3aj} g^{\gamma})^{d_i} (\mathrm{mod} n), \quad 1 \leq i \leq r$$

and thus,

$$\frac{\prod_{i=1}^{r} W_i}{\hat{S}^{a3j} G^{\gamma}} = \frac{M^{3ja \sum_{i=1}^{r} d_i} g^{\gamma \sum_{i=1}^{r} d_i}}{\hat{S}^{a3j} G^{\gamma}} = \left(\frac{S}{\hat{S}}\right)^{a3j} = T \ (\mathrm{mod} n)$$

1) On the first ℓ rounds, $j = 0$ or $j = 1$ if and only if $T = 1$ or $T \neq 1$ respectively. Among them j must be 1 at least once, i.e., the cosigner can obtain $\overline{T} = \left(\frac{S}{\hat{S}}\right)^{a3} (\neq 1) (\bmod n)$.

2) On the last $(\ell+1)$-th round, $j \in \{0, 1, 2, \ldots, k\}$, the cosigner can test $\overline{T}^{\overline{j}} =? T (\bmod n)$ through trial and error for $\overline{j} \in \{0, 1, 2, \ldots, k\}$. Thus, the verifier checks whether $\overline{j} = j$ holds for all $(\ell + 1)$ rounds. If so, then V always rejects $\hat{S}$.

(2) *Soundness property*

If $\hat{S} = S$, then $\frac{\prod_{i=1}^{r} W_i}{\hat{S}^{a3j} G^{\gamma}} = \frac{M^{3ja \sum_{i=1}^{r} d_i} g^{\gamma \sum_{i=1}^{r} d_i}}{\hat{S}^{a3j} G^{\gamma}} = \left(\frac{S}{\hat{S}}\right)^{a3j} = 1 \ (\mathrm{mod} n)$ is always true.

In each run, the best strategy for the cosigner to obtain j is through guessing. Therefore, the probability to successfully cheat the verifier (which is to reject a valid signature) is $\frac{1}{2^{\ell}} \times \frac{1}{k+1}$.

(3) *Non-transferability*

It is clear that the verifier V can always simulate and generate a valid transcript since the disavowal is zero-knowledge. Any verifier interacting with cosigner p_i, $1 \leq i \leq r$ learns no information from cosigner p_i. Thus the verifier V is unable to convince any other party V' the invalidity of the signature without the cooperation of all cosigners p_i.

4.6. **A brief example for practical discussion.** In this section, we have pointed out that the proposed scheme is very helpful to solve the problem of designing a robust multi-authority electronic voting (e-voting) system.

It is strongly suggested by many cryptographic experts that a significant solution for a robust electronic voting (e-voting) system to prevent cheating from the malicious authority is the design of a system consisting of more than one authority (so that the authorities have little chance to cheat unless they conspire all together). However, in a multi-authority e-voting system, a blind multi-signature is needed for the authorities to cosign a blank vote in order to avoid the system from linking the votes with the voters. Further, if the blind multi-signature made by the authorities is undeniable as well, the system can convince any verifier to accept valid votes as well as reject invalided votes included in the final tally.

Since the proposed multi-signature scheme is not only a blind signature but also an undeniable signature in which the modulus used by each cosigner is identical, it is an attractive candidate in the design of a robust e-voting system.

5. Security and Complexity Analysis.

In 2001, Okamoto and Pointcheval proposed "The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes" [66]. In their proposal, they showed how the gap problems find natural applications in cryptography, not only for proving the security of very efficient schemes, but also for solving the Chaum's undeniable signature.

5.1. Provable security analysis.

5.1.1. *The Gap Diffie-Hellman problem.* Refer to [66], let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ be any relation. The inverting problem of f is the classical computational version, and by the R-decision problem of f a generalization of the decision problem is introduced, for any relation

$$R : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\},$$

– *the inverting problem* of f is, given x, to compute any y such as $f(x,y) = 1$ if it exists, or to answer **Fail**.
– *the R-decision problem* of f is, given (x,y), to decide whether $R(f,x,y) = 1$ or not.

The Gap problem of f is to solve the inverting problem of f with the help of the oracle of the decision problem of f.

Review Diffie-Hellman Problems [78] with its gap variations. Consider a group G of prime order q, g is a generator of G. Then we give three problems as follows:
– *The Inverting Diffie-Hellman Problem* (C-DH) (i.e., the Computational Diffie-Hellman problem): given a triple of G elements (g, g^a, g^b), find the element $C = g^{ab}$.
– *The Decision Diffie-Hellman Problem* (D-DH): given a quadruple of G elements $(g, g^a, g^b,$ $g^c)$, decide whether $c = ab \bmod q$ or not.
– *The Gap-Diffie-Hellman Problem* (G-DH): given a triple (g, g^a, g^b), find the element $C = g^{ab}$ with the help of a Decision Diffie-Hellman Oracle.

Note that the decision problem is the default one, once the relation f is defined by

$$f((g,A,B),C) \underline{\underline{\text{def}}} \log_g C =? \log_g A \times \log_g B \ (\text{mod} q)$$

which is a priori not a polynomially computable function. Furthermore, the weil pairing scheme on elliptic curve is an easy case of gap problem.

In weakly undeniable signatures, contrarily to plain signatures, the verification process must be intractable without the help of the signer. And therefore, the confirmer (which can be the signer himself) can be seen as a decision oracle. We will prove that the full-domain hash [75] variant of our scheme is secure under the Gap-DH problem, in the random oracle model [79].

Description. Our proposal consists of a interactive signature process and an interactive confirmation protocol.
– Setting:

1) The KGC generates and distributes a pair of public/secret keys (e_i, d_i) and n to each cosigner P_i, $1 \le i \le r$ satisfying $\sum_{i=1}^{r} e_i d_i \equiv 1 \pmod{\varphi(n)}$. The secret key of the cosigner is $d_i \in Z_n^*$ while his public key is e_i.
2) The KGC chooses a random value g is a generator of the largest multiplicative cyclic subgroup of Z_n^* with order $\frac{\varphi(n)}{2}$ and computes $G_i \equiv g^{(e_i+1)d_i} \pmod{n}$, $1 \le i \le r$, and $G \equiv g^d \pmod{n}$, where $d = \sum_{i=1}^{r} d_i$.
3) The KGC determines a full-domain hash function $H : \{0, 1\}^* \to C$

– Signature of M: in order to sign a message M, the cosigner computes and returns $S = M^d(\mathrm{mod} n)$ where $d = \sum_{i=1}^{r} d_i$.

– Confirmation/Disavowal of (M, S): an interactive proof is used to convince the any third verifier V whether

$$\log_M S \equiv \log_g G(\mathrm{mod} n) \text{ and } \sum_{i=1}^{r} e_i d_i' = \sum_{i=1}^{r} e_i d_i = 1(\mathrm{mod}\varphi(n)).$$

We use the classical full-domain hash technique [75]. If this hash function H is furthermore assumed to behave like a random oracle [79,80], this scheme can be proven secure.

5.1.2. *Unforgeability of the proposed scheme.* The theorem below proves that the signature security is equivalent to solving the Gap Diffie-Hellman problem. Since D-DH is easy in our scheme, our signature security is also equivalent to solving the C-DH problem.

Theorem 5.1. *An existential forgery under adaptively chosen-message attacks is equivalent to the Gap Diffie-Hellman problem.*

Proof: For this equivalence, we can easily see that if we can break the G-DH (g, G_i, M) problem, possibly with access to a D-DH $(g, G_i, M, M^{(e_i+1)\hat{d}_i})$ oracle, for randomly choosing $\hat{d}_i \in [1, \frac{\varphi(n)}{2})$, then we can forge a signature in a universal way: first, a C-DH (g, G_i, M) oracle is simulated (with overwhelming probability) by the confirmation/disavowal protocols. Then, for any message$m$, we can compute $H(m) = M$ and C-DH (g, G_i, M) $i = 1, 2, \dots, r$. Let $C_i =$ C-DH $(g, G_i, M) = M^{(e_i+1)d_i}$, $i = 1, 2, \dots, r$, and then we can obtain a forgery signature $S = \left(\prod_{i=1}^{r} C_i\right) M^{-1} = M^{\sum_{i=1}^{r} d_i}$.

Therefore, the security of this weakly undeniable signature scheme is weaker than the G-DH (g, G_i, M) problem.

Conversely, we can use the same techniques as in [75,80] for the security of the full-domain hash signature. Assume we have r independent signing oracles O_i, $i = 1, 2, \dots, r$ and a full-domain hash oracle. Suppose that there is an existential forgery with probability ε within time t after q_i average number of queries to the signing oracle O_i, $i = 1, 2, \dots, r$. We can use $q_M + 1 = \sum_{i=1}^{r} q_i$ to indicate the total of each average number of queries times required to complete rO_i's. Now, given G-DH (g, g^a, g^b), where g is the generator of C (C is the longest cyclic subgroup of Z_n^* with order $\frac{\varphi(n)}{2}$), a, $b \in [1, \frac{\varphi(n)}{2})$, we try to find the element $g^{ab} \in C$ with the help of a D-DH (g, g^a, g^b, g^c) oracle. Since the secret keys of r cosigners d_i, $i = 1, 2, \dots, r$ are randomly selected, $d = \sum_{i=1}^{r} d_i$ is also random. Thus, we may assume that $d = a(\mathrm{mod}\frac{\varphi(n)}{2})$; H is a full-domain function onto C, then we can let $g^b = M^\tau = H(m)$ for some $\tau \in [1, \frac{\varphi(n)}{2})$ under adaptively chosen-message m. G-DH (g, g^a, g^b) can therefore be regarded as G-DH (g, G, M). Thus, we are able to simulate any interaction with the adversary in an indistinguishable setting from a real attack.

– Confirmation/disavowal queries are perfectly simulated by simulating the appropriate proofs correctly chosen by the D-DH $(g, G_i, M, M^{(e_i+1)\hat{d}_i})$ oracle, for $i = 1, 2, \ldots, r$.

– Any hash query message m is answered in a probabilistic way. Moreover, we choose a corresponding random exponent $\tau \in [1, \frac{\varphi(n)}{2})$ such that $H(m) = M^\tau$, and in this case, we may assume that the probability is p. Otherwise, we set $H(m) = g^\tau$, and the simulation aborts.

– Any signing query message m (assumed to have already been asked to H) is answered as follows: if $H(m)$ is defined, then $S_i = H(m)^{(e_i+1)d_i}$ for the i-th signing oracle O_i, $i = 1, 2, \ldots, r$.

Finally, the adversary outputs a forgery S for an adaptively chosen-message m (also assumed to have been asked to H) such that it satisfies $H(m) = M^\tau$ with probability p, then we have $S_i = g^{b\tau(e_i+1)d_i}, i = 1, 2, \ldots, r$. Therefore, we obtain a forgery signature

$$S = \prod_{i=1}^{r} S_i M^{-\tau} = \prod_{i=1}^{r} g^{b\tau(e_i+1)d_i} M^{-\tau} = g^{b\tau(d+1)} M^{-\tau} = M^{\tau d}.$$

Consequently, $S^{\frac{1}{\tau}} = M^d = \text{C-DH}(g, G, M) = \text{C-DH}(g, g^a, g^b) = g^{ab} (\text{mod} n)$.

The success probability is exactly the same as for the full-domain hash technique [75]. $\varepsilon' = \varepsilon(1-p)^{q_M} \times p \geq \frac{1}{2}\exp(-1) \times \frac{\varepsilon}{q_M}$, for the average number of queries times $q_M \geq 1$, while $p = \frac{1}{q_M+1}$.

$$\begin{aligned}
//\because\ & \log 2 + \frac{1}{p}\log(1-p) \\
= & \left[1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + (-1)^{n+1}\frac{1}{n} + \cdots\right] \\
& + \frac{1}{p}\left[-p - \frac{p^2}{2} - \frac{p^3}{3} - \frac{p^4}{4} - \cdots - \frac{p^n}{n} - \cdots\right] \\
= & \frac{-1-p}{2} + \left(\frac{1}{3} - \frac{1}{4}\right) + \left(\frac{1}{5} - \frac{1}{6}\right) + \left(\frac{1}{7} - \frac{1}{8}\right) + \left(\frac{1}{9} - \frac{1}{10}\right) \\
& + \left(\frac{1}{11} - \frac{1}{12} + \frac{1}{13} - \frac{1}{14} + \cdots\right) + \frac{1}{p}\left(-\frac{p^3}{3} - \frac{p^4}{4} - \frac{p^5}{5}\right) \\
& + \left(-\frac{p^5}{6} - \frac{p^6}{7} - \cdots - \frac{p^{n-1}}{n} - \cdots\right) \\
\geq & -1 + \frac{1}{12} + \frac{1}{30} + \frac{1}{56} + \frac{1}{90} + K - \frac{1}{3 \times 2^2} - \frac{1}{4 \times 2^3} - \frac{1}{5 \times 2^4} \\
& - \left(\frac{1}{6 \times 2^5} + \frac{1}{7 \times 2^6} + \frac{1}{8 \times 2^7} + \cdots + \frac{1}{n \times 2^{n-1}} + \cdots\right) \\
& \text{for } 0 \leq p \leq \frac{1}{2}, \quad K > 0 \\
\geq & -1 + \frac{1}{90} - \left(\frac{1}{6 \times 2^5} + \frac{1}{6 \times 2^6} + \frac{1}{6 \times 2^7} + \cdots + \frac{1}{6 \times 2^{n-1}} + \cdots\right) \\
\geq & -1
\end{aligned}$$

$$\therefore \log 2 + \frac{1}{p}\log(1-p) \geq -1$$

$$\Rightarrow \log 2(1-p)^{\frac{1}{p}} \geq -1$$

$$\Rightarrow \varepsilon' = \varepsilon(1-p)^{q_M} \times p = \varepsilon(1-p)^{q_M+1} \times \frac{p}{1-p}$$

$$= (1-p)^{\frac{1}{p}}\frac{\varepsilon}{q_M} \geq \frac{1}{2}\exp(-1) \times \frac{\varepsilon}{q_M} \;//$$

5.2. **Complexity analysis.** Let $T_{\exp}$ and T_{mul} denote the execution time for an exponential modulo n operation and a multiplication modulo n operation respectively. Suppose that the values of $g^{e_i+1}(\bmod n)$, $1 \leq i \leq r$, $g^{-1}(\bmod n)$, $G^{-1}(\bmod n)$, $gG(\bmod n)$ and $(gG)^{-1} \bmod n$ have been precomputed. Then the computational complexity of the proposed scheme can be analyzed as follows.

5.2.1. *Time complexity of signature generation of Version 1.* It is easy to see that in the blinding and requesting steps of signature generating phase of Version 1, the requester U needs $2(T_{\exp} + rT_{mul})$ for blinding m into (B_{1i}, B_{2i}), $1 \leq i \leq r$. In the signing step, each participant cosigner P_i, $1 \leq i \leq r$, takes $2T_{\exp}$ to compute his blindly partial signature (w_{1i}, w_{2i}) and takes $rT_{\exp}$ to store the blind message $B_{1i}(g^{e_i+1})^{-1} \bmod n \equiv M^a$. Totally, $r(2T_{\exp} + T_{mul})$ is required in this step. In the step of verifying partial signatures, U needs $r(3T_{\exp} + T_{mul})$ time for computing the values of w_{1i}^b, w_{2i}^a, $w_i \equiv w_{1i}^b w_{2i}^a(\bmod n)$ and checking the equality of $w_i \equiv G_i^{a+b}(\bmod n)$, $1 \leq i \leq r$. Finally, in the producing multi-signature step, U needs $1T_{\exp} + (2r+2)T_{mul}$ for computing the values of W_1, W_2, S_1, S_2 and multi-signature S. Consequently, the total time required in the signature generating phase of Version 1 is $(2 + 2r + 3r + 1)T_{\exp} + (6r + 2)T_{mul}$.

5.2.2. *Time complexity of signature generation of Version 2.* The time complexity of signature generation of Version 2 can be analyzed in a similar way as that of Version 1. However, we observe that the computations needed only in Version 1 include w_{1i}^b, w_{2i}^a, $w_i \equiv w_{1i}^b w_{2i}^a$ and $G_i^{a+b}(\bmod n)$, $1 \leq i \leq r$, while the computations that are additionally needed in Version 2 include $W \equiv W_1^b W_2^a(\bmod n)$ and $W \equiv (gG)^{a+b}(\bmod n)$. Consequently, in comparison with Version 1, the time complexity of signature generation of Version 2 is decreased by $(2r+6)T_{\exp} + (5r+3)T_{mul}$. This is because all partial signatures are verified as a whole by the requester.

5.2.3. *Time complexity of confirmation protocol.* In the confirmation protocol, the verifier needs $2T_{\exp} + 1T_{mul}$ for computing r challenges U_i's, $1 \leq i \leq r$; $(1T_{\exp} + 1T_{mul})$ for computing $\{W_i\}_{i=1}^r$, $1T_{\exp}$ for computing $\overline{W}_i$. On the other hand, each participant cosigner needs $(1T_{\exp} + 1T_{mul})$ and one $T_{\exp}$ for computing his response W_i and $\overline{W_i}$, $1 \leq i \leq r$ respectively; also needs $2T_{\exp}$ and $1T_{mul}$ for reconstructing the U_i, $1 \leq i \leq r$. $(1T_{\exp} + 1T_{mul})$ for reconstructing the $\{W_i\}_{i=1}^r$, $2T_{\exp} + 1T_{mul}$ for computing $\left\{\overline{\overline{W_i}}\right\}_{i=1}^r$; accordingly, in the verification step, V needs $1T_{\exp} + (r-1)T_{mul}$. Finally, it requires totally $(2r + 2r + 2r + 3r + 1)T_{\exp} + (r + r + r + 3r - 1)T_{mul} = (9r+1)T_{\exp} + (6r-1)T_{mul}$ for the confirmation protocol.

5.2.4. *Time complexity of disavowal protocol.* Similarly, in the disavowal protocol, the verifier needs $4rT_{\exp} + 2rT_{mul}$ for computing r challenges $\left\{M^{a3j}g^\gamma, \hat{S}^{a3j}G^\gamma\right\}$'s, $1 \leq i \leq r \ldots$. A total amount of $4rT_{\exp} + 2rT_{mul}$ is also needed for all cosigners p_i $(1 \leq i \leq r)$ to reconstruct the step $(M^{a3j}g^\gamma) \cdot g^{-\gamma} \equiv M^{a3}$. Accordingly, it requires totally $8rT_{\exp} + 4rT_{mul}$ time for the disavowal protocol.

Tables 1-4 below depict the time complexity of the signature generation in Version 1 and Version 2 as well as the time complexity of the confirmation and disavowal protocol.

Table 1. Time complexity of signature generation of Version 1

Steps \ operations	T_{exp}	T_{mul}
Step 1. (Blinding and Requesting)	2	$2r$
Step 2. (Signing)	$2r$	r
Step 3. (Verifying each individual partial signature)	$3r$	r
Step 4. (Generating multi-signature)	1	$2r+2$
Total	$5r+3$	$6r+2$

Table 2. Time complexity of signature generation of Version 2

Steps \ operations	T_{exp}	T_{mul}
Step 1. (Blinding and Requesting)	2	$2r$
Step 2. (Signing)	$2r$	r
Step 3. (Verifying each individual partial signature)	3	$2r-1$
Step 4. (Generating multi-signature)	1	4
Total	$2r+6$	$5r+3$

Table 3. Time complexity of confirmation protocol

Steps \ operations	T_{exp}	T_{mul}
Step 1. [V's turn]	$2r$	r
Step 2. [P_i's turn, $1 \leq i \leq r$]	$2r$	r
Step 3. [V's turn]	/	/
Step 4. [P_i's turn, $1 \leq i \leq r$]	$2r$	r
Step 5. [V's turn]	$3r+1$	$3r-1$
Total	$9r+1$	$6r-1$

6. **Conclusions.** In this paper, we have introduced the concept of an integrated signature and proposed a new integrated signature scheme that has the following engaging properties:

(1) It is not only a blind signature but a weakly undeniable signature as well.
(2) It is a distributed RSA-type multi-signature.
(3) The modulus used by each participant cosigner is identical.
(4) It is identical in form to a standard RSA signature.
(5) The length of the signature is unrelated to the number of participant cosigners.
(6) "Zero-knowledge undeniable signatures" is applied to execute the confirmation/disavowal protocols.

Table 4. Time complexity of disavowal protocol

operations Steps	T_{exp}	T_{mul}
Step 1. [V's turn]	$4r$	$2r$
Step 2. [P_i's turn, $1 \leq i \leq r$]	blob $(\delta, \overline{j})$	/
Step 3. [V's turn]	/	/
Step 4. [P_i's turn, $1 \leq i \leq r$]	$4r$	$2r$
Step 5. [V's turn]	open blob$(\delta, \overline{j})$	/
Total	$8r$	$4r$

(7) "The Gap Diffie-Hellman Problem" is used to solve the forgery problems.

Ever since Chaum proposed the first single-signer undeniable signature technique in 1989, many researchers developed one-vector variants and generalized undeniable signature schemes based on his theorem, such as partial blindness, fair blindness, convertible undeniable signatures, zero-knowledge undeniable signatures, designated confirmer signature, identity based undeniable signatures, certificateless undeniable signature scheme [31-33,39,51]. We are the first to propose the multi-signer distributed undeniable signature technique, and similar to the single-signer undeniable signature scheme, we aim to develop multi-signer variants and generalized distributed undeniable signatures for our future work. Furthermore, in regards to our research work, designing and implementing a robust large-scale engineering tender system can be made possible by promoting our signature to become a "fair" blind and weakly undeniable multi-signature.

REFERENCES

[1] D. Chaum, Blind signatures for untraceable payments, *Proc. of Crypto.*, pp.199-203, 1982.

[2] A. K. Awasthi and S. Lal, Proxy blind signature scheme, *Transactions on Cryptology*, vol.2, no.1, pp.5-11, 2005.

[3] A. Boldyreva, Threshold signatures, multi-signatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme, *Public Key Cryptography, LNCS*, vol.2567, pp.31-46, 2003.

[4] G. Bassard, D. Chaum and C. Cr'epeau, Minimum disclosure proofs of knowledge, *Journal of Computer and System Sciences*, vol.37, pp.156-189, 1988.

[5] J. Camenisch, M. Koprowski and B. Warinschi, Efficient blind signatures without random oracles, *Security in Communication Networks, LNCS*, vol.3352, pp.134-148, 2005.

[6] J. L. Camenisch, J. M. Piveteau and M. A. Stadler, Blind signatures based on the discrete logarithm problem, *Proc. of Eurocrypt*, pp.428-432, 1994.

[7] H. Y. Chien, J. K. Jan and Y. M. Tseng, RSA-based partially blind signature with low computation, *Proc. of ICPADS*, pp.385-389, 2001.

[8] D. N. Duc, J. H. Cheon and K. Kim, A forward-secure blind signature scheme based on the strong RSA assumption, *Information and Communications Security, LNCS*, vol.2836, pp.11-21, 2003.

[9] P. Horster, M. Michels and H. Petersen, Meta message recovery and meta blind signature schemes based on the discrete logarithm problem and their applications, *Proc. of Asiacrypt*, pp.234-237, 1994.

[10] S. Miyazaki and K. Sakurai, A practical off-line digital money system with partially blind signatures based on the discrete logarithm problem, *IEICE Trans. Fundamentals*, vol.E83-A, no.1, pp.106-108, 2000.

[11] D. Pointcheval and J. Stern, Provably secure blind signature schemes, *Proc. of Asiacrypt*, pp.252-265, 1996.

[12] R. L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public key cryptosystem, *CACM*, vol.21, no.2, pp.120-126, 1978.

[13] M. Stadler, J. M. Piveteau and J. Camenisch, Fair blind signatures, *Proc. of Eurocrypt*, pp.209-219, 1995.

[14] F. Zhang and K. Kim, ID-based blind signature and ring signature from pairings, *Proc. of Asiacrypt, LNCS*, vol.2501, pp.533-547, 2002.

[15] F. Zhang, R. Safavi-Naini and W. Susilo, Efficient verifiably encrypted signature and partially blind signature from bilinear pairings, *Indocrypt, LNCS*, vol.2904, pp.191-204, 2003.

[16] M. Abe and E. Fujisaki, How to date blind signatures, *Advances in Cryptology – Asiacrypt, LNCS*, vol.1163, pp.244-251, 1996.

[17] F. Zhang and K. Kim, Efficient ID-based blind signature and proxy signature from bilinear pairings, *Proc. of ACISP, LNCS*, vol.2727, pp.312-323, 2003.

[18] A. Awasthi and S. Lal, Proxy blind signature scheme, *Transcations on Cryptology*, vol.2, no.1, pp.5-11, 2005.

[19] D. Chaum, B. den Boer, E. van Heyst, S. Mjoelsnes and A. Steenbeek, Efficient offline electronic checks, *Proc. of Eurocrypt*, pp.294-301, 1989.

[20] N. Ferguson, Single term off-line coins, *Proc. of Eurocrypt*, pp.318-328, 1993.

[21] A. Fujioka, T. Okamoto and K. Ohta, A practical secret voting scheme for large scale election, *Proc. of Auscrypt*, pp.244-251, 1992.

[22] S. Kim and H. Oh, A new electronic check system with reusable refunds, *International Journal of Information Security*, vol.1, no.3, pp.175-188, 2002.

[23] C. L. Lin, H. F. Lin and C. Y. Chen, An extended RSA based multiauthority electronic voting system, *Proc. of the 10th National Conference on Information Security*, Taiwan, pp.139-148, 2000.

[24] T. Okamoto and K. Ohta, Universal electronic cash, *Proc. of Crypto.*, pp.324-337, 1991.

[25] S. H. Yun and S. J. Lee, An electronic voting scheme based on undeniable blind signature scheme, *Proc. of the 37th IEEE Carnahan Conference on Security*, pp.163-167, 2003.

[26] T. Balopoulos, S. Gritzalis and S. K. Katsikas, Specifying privacy-preserving protocols in typed MSR, *Computer Standards & Interfaces*, vol.27, no.5, pp.501-512, 2005.

[27] Z. Huang, K. Chen and W. Kou, Untraceable partially blind signature based on DLOG problem, *Journal of Zhejiang University (Science)*, vol.5, no.1, pp.40-44, 2004.

[28] F. Yang and J. Jan, A provably secure scheme for restrictive partially blind signatures, *Cryptology ePrint Archive*, 2004.

[29] S. Chow, L. Hui, S. Yiu and K. Chow, Two improved partially blind signature schemes from bilinear pairings, *Cryptology ePrint Archive*, 2004.

[30] D. Chaum and H. Van Antwerpen, Undeniable signatures, *Advances in Cryptology, LNCS*, vol.435, pp.212-216, 1989.

[31] J. Boyar, D. Chaum, I. Damgard and T. Pedersen, Convertible undeniable signatures, *Proc. of Crypto.*, pp.189-205, 1990.

[32] D. Chaum, Zero-knowledge undeniable signatures, *Proc. of Eurocrypt*, pp.458-464, 1990.

[33] D. Chaum, Designated confirmer signature, *Proc. of Eurocrypt*, pp.86-91, 1994.

[34] D. Chaum, A. Fiat and M. Naor, Untraceable electronic cash, *Proc. of Crypto.*, pp.319-327, 1988.

[35] R. Gennaro, H. Krawczyk and T. Rabin, RSA-based undeniable signatures, *Proc. of Crypto.*, pp.132-149, 1997.

[36] M. Jakobsson, K. Sako and R. Impagliazzo, Designated verifier proof and their applications, *Proc. of Eurocrypt*, pp.143-154, 1996.

[37] K. Kurosawa and S. H. Heng, 3-move undeniable signature scheme, *Eurocrypt, LNCS*, vol.3494, pp.181-197, 2005.

[38] F. Laguillaumie and D. Vergnaud, Time-selective convertible undeniable signatures, *Topics in Cryptology – CT-RSA, LNCS*, vol.3376, pp.154-171, 2005.

[39] B. Libert and J. J. Quisquater, Identity based undeniable signatures, *Topics in Cryptology – CT-RSA, LNCS*, vol.2964, pp.112-125, 2004.

[40] J. Monnerat and S. Vaudenay, Undeniable signatures based on characters: How to sign with one bit, *Public Key Cryptography, LNCS*, vol.2947, pp.69-85, 2004.

[41] J. Monnerat and S. Vaudenay, Generic homomorphic undeniable signatures, *Asiacrypt, LNCS*, vol.3329, pp.354-371, 2004.

[42] S. D. Galbraith and W. Mao, Invisibility and anonymity of undeniable and confirmer signatures, *CT-RSA, LNCS*, vol.2612, pp.80-97, 2003.

[43] G. Wang, J. Zhou and R. H. Deng, Cryptanalysis of the Lee-Hwang group-oriented undeniable signature schemes, *Cryptology ePrint Archive*, 2002.

[44] B. Libert and J. Quisquater, ID-based undeniable signatures, *Advances in CT-RSA, LNCS*, vol.2964, pp.112-125, 2004.

[45] A. Koide, R. Tso, T. Okamoto and E. Okamoto, A restricted undeniable designated verifier signature, *APSCC*, pp.1375-1380, 2008.

[46] M. Jakobsson, Blackmailing using undeniable signatures, *Proc. of Eurocrypt*, pp.425-427, 1994.

[47] T. Nakanishi, H. Watanabe and T. Fujiwara, Anonymous auction protocol using undeniable signature, *Proc. of Symposium on Cryptography and Information Security*, 1995.

[48] S. D. Galbraith, W. Mao and K. G. Paterson, RSA-based undeniable signatures for general moduli, *Topics in Cryptology – CT-RSA, LNCS*, vol.2271, pp.200-217, 2002.

[49] T. Miyazaki, An improved scheme of the gennaro-krawczyk-rabin undeniable signature system based on RSA, *ICISC, LNCS*, vol.2015, pp.135-149, 2000.

[50] F. Laguillaumie and D. Vergnaud, Multi-designated verifiers signatures, *Information and Communications Security, LNCS*, vol.3269, pp.495-507, 2004.

[51] S. Duan, Certificateless undeniable signature scheme, *Information Sciences*, vol.178, pp.742-755, 2008.

[52] L. Wang, Z. Cao, X. Li and H. Qian, Simulatability and security of certificate less threshold signatures, *Information Science*, vol.177, pp.1382-1394, 2007.

[53] R. W. Zhu, G. Yang and D. S. Wong, An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices, *Theoretical Computer Science*, vol.378, pp.198-207, 2007.

[54] L. E. Aimani and D. Vergnaud, Gradually convertible undeniable signatures, *ACNS, LNCS*, vol.4521, pp.478-496, 2007.

[55] L. E. Aimani, Toward a generic construction of universally convertible undeniable signatures from pairing-based signatures, *Indocrypt, LNCS*, vol.5365, pp.145-157, 2008.

[56] L. E. Aimani, Toward a generic construction of convertible undeniable signatures from pairing-based signatures, *Cryptology ePrint Archive, Report 2009/362*, 2009.

[57] J. C. N. Schuldt and K. Matsuura, An efficient convertible undeniable signature scheme with delegatable verification, *Information Security, Practice and Experience, LNCS*, vol.6047, pp.276-293, 2010.

[58] K. Ohta and T. Okamoto, A digital multi-signature scheme based on the Fiat-Shamir scheme, *Proc. of Asiacrypt*, pp.139-148, 1991.

[59] Y. S. Chang, T. C. Wu and S. C. Huang, ElGamal-like digital signature and multi-signature schemes using self-certified public keys, *Journal of Systems and Software*, vol.50, no.2, pp.99-105, 2000.

[60] T. Hardjono and Y. Zheng, A practical digital multi-signature scheme based on discrete logarithms, *Proc. of Auscrypto*, pp.122-132, 1992.

[61] T. Okamoto, A digital multi-signature scheme using bijective public-key cryptosystems, *ACM Trans. on Comp. Sys.*, vol.6, no.8, pp.432-441, 1988.

[62] S. F. Pon, E. H. Lu and J. Y. Lee, Dynamic reblocking RSA-based multi-signatures scheme for computer and communication networks, *IEEE Communications Letters*, vol.6, no.1, pp.43-44, 2002.

[63] V. Shoup, Practical threshold signatures, *Proc. of Eurocrypt, LNCS*, vol.1807, pp.207-220, 2000.

[64] M. Tada, An order-specified multi-signature scheme secure against active insider attacks, *Information Security and Privacy, LNCS*, vol.2384, pp.328-345, 2002.

[65] S. H. Yun and H. S. Lim, The convertible undeniable multi-signature scheme suitable for digital copyright protection, *ICHIT*, pp.594-599, 2008.

[66] T. Okamoto and D. Pointcheval, The gap-problems: A new class of problems for the security of cryptographic schemes, *PKC, LNCS*, vol.1992, pp.104-118, 2001.

[67] H. F. Lin and C. Y. Chen, An extended RSA based generalized group-oriented signature scheme, *Proc. of the 10th National Conference on Information Security*, Hwalein, Taiwan, 2000.

[68] K. H. Rosen, *Elementary Number Theory and Its Applications*, 2nd Edition, Addison-Wesley Publishing Company, 1987.

[69] C. Y. Chen, H. F. Lin and C. C. Chang, An efficient generalized group-oriented signature scheme, *International Journal of Innovative Computing, Information and Control*, vol.4, no.6, pp.1335-1345, 2008.

[70] J. L. Lin, H. F. Lin, C. Y. Chen and C. C. Chang, A multiauthority electronic voting protocol based upon a blind multi-signature scheme, *IJCSNS*, vol.6, no.12, pp.266-274, 2006.

[71] J. Gordan, Strong RSA key, *Electronics Letters*, vol.20, pp.514-516, 1984.

[72] C. S. Lai, W. C. Yang and C. H. Chen, Efficient method for generating strong primes with constraint of bit length, *Electronics Letters*, vol.27, no.20, pp.1808-1808, 1991.

[73] K. H. Rosen, *Elementary Number Theory and Its Applications*, 2nd Edition, Addison-Wesley Publishing Company, 1987.

[74] G. T. Simmons and M. J. Norries, Preliminary comments on the M.I.T. public-key cryptosystem, *Cryptologia*, vol.1, pp.406-414, 1977.

[75] J. S. Corone, On the exact security of full domain hash, *Crypto, LNCS*, vol.1880, pp.229-235, 2000.

[76] N. Y. Lee, T. Hwang and C. H. Wang, The security of two ID-based multi-signature protocols for sequential and broadcasting architectures, *Information Proceeding Letters*, vol.70, no.2, pp.79-81, 1999.

[77] T. C. Wu, S. L. Chou and T. S. Wu, Two ID-based multi-signature protocols for sequential and broadcasting architectures, *Computer Communications*, vol.19, pp.851-856, 1996.

[78] W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol.22, no.6, pp.644-654, 1976.

[79] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for de-signing efficient protocols, *Proc. of the 1st CCS*, New York, pp.62-73, 1993.

[80] M. Bellare and P. Rogaway, The exact security of digital signatures – How to sign with RSA and Rabin, *Eurocrypt, LNCS*, vol.1070, pp.399-416, 1996.