A New Type of Designated Confirmer Signatures for a Group of Individuals

Baodian Wei[1,3], Fangguo Zhang[1,3], and Xiaofeng Chen[2,3]
(Corresponding author: Baodian Wei)
Department of Electronics and Communication Engineering,[1]
Sun Yat-sen University Guangzhou, 510275, China
Department of Computer Science, Sun Yat-sen University, Guangzhou, 510275, China[2]
Guangdong Key Laboratory of Information Security Technology, Guangzhou, 510275, China[3]
(Email: weibd@mail.sysu.edu.cn)
(Received Jan. 15, 2007; revised and accepted Aug. 2, 2007)

Abstract

A new concept of society-oriented designated confirmer signatures (SDCS) is introduced in this paper. SDCS is well suited to applications where the capability of the signer and that of the confirmer are both expected to be shared among a group of individuals. The ways to share the signing capability and the confirming capability are different due to the distinct stabilities of the signer group and the confirmer group. Based on the techniques of threshold cryptography, a concrete SDCS scheme is proposed and its security is analyzed. Ordinary designated confirmer signatures and threshold designated confirmer signatures can be regarded as special cases of the proposed society-oriented designated confirmer signatures. Meanwhile, our scheme can be converted into an ordinary signature scheme or a designated verifier signature scheme.

Keywords: Designated confirmer signature, digital signature, society-oriented designated confirmer signature, threshold cryptography

1 Introduction

Digital signatures [6] are used in the open distributed systems to guarantee the authenticity of data. The nonrepudiation property is achieved by providing universal verifiability for the conventional signatures. This is a property suitable for such situations as the dissemination of announcements or public keys, where the more copies distributed the better. However, this property is not always desirable. Sometimes the signer wishes that the recipient of the signature would not be able to present it to any other party at will. Undeniable signatures introduced by Chaum and van Antwerpen [4], were proposed as a solution to this issue where the signature can not be verified without collaboration of the legitimate signer. After their initial work, various undeniable signature schemes have been proposed [1, 5, 9, 11, 13, 19].

One limitation in these undeniable signatures is that valid signatures would no longer be verifiable if the signer is absent or unwilling to cooperate. The notion of the designated confirmer signature (DCS) was proposed by Chaum [3] to solve this weakness of undeniable signatures. In this situation, the ability to verify signatures is delegated to a third party, namely, the confirmer. The confirmer, previously designated by the signer, is able to either confirm or disavow the validity of a signature but not able to forge any signature. Since its invention, several concrete realizations of designated confirmer signatures were presented [2, 7, 8, 15, 16].

Consider the scenario where the capability of the signer and that of the confirmer are both needed to be shared among groups of individuals. For example, in a big software company where all kinds of signatures on digital products are signed coordinately by several directors rather than just by the chairman of the board. On the other hand, to make the customers more convinced, the signatures are suggested to be confirmed by more than one agent. Customers can appeal to agents they are familiar with. Also can a customer appeal to other agents if he does not trust the original agents any longer. It implies that the stability of the signer group and that of the confirmer group are quite different. Accordingly, the ways to share the signing capability and the confirming capability are different. A fixed private key may be shared by members of the signer group while there should not be a similar key among the confirmer group. Ordinary DCS can not solve this problem. In this paper, we introduce the concept of society-oriented designated confirmer signature (SDCS) as a solution. Based on the techniques of threshold cryptography, a concrete SDCS scheme is proposed and its security is analyzed. Ordinary DCS and threshold DCS can be regarded as special instances of the proposed SDCS. In the meantime, our scheme can be converted into an ordinary signature scheme or a designated

verifier signature scheme.

1.1 Related Works

The first realization of a DCS was presented in [3]. In their scheme, the DCS signature σ on the message m is constructed as a triple (a, b, α), where $a = g^r, b = PK_C^r$, and the RSA signature $\alpha = (F(a, b) \oplus H(m))^{SK_S}$. Since both the inverse function F and the hash function H are public, the RSA signature is forgeable. Therefore, another procedure to verify if $b = a^{SK_C}$ holds is necessary. Obviously, only the confirmer can initiate this procedure. Forging an RSA signature by calculating $(a, b) = F^{-1}(\alpha^{PK_S} \oplus H(m))$ with the previously fixed α can not guarantee the specific structural property of $b = a^{SK_C}$.

Okamoto provided in [16] the first formal definition for a DCS scheme and showed constructively that a DCS scheme was equivalent to a public-key encryption scheme with respect to existence. A straightforward way to construct a DCS scheme was suggested by Okamoto: firstly, the message was signed by using an ordinary signature scheme, then the signature was encrypted by using the designated confirmer's public key and finally, the resulting ciphertext would serve as the DCS signature. General zero-knowledge proofs for NP statements should be used in such constructions and cannot really be used in practice. Two concrete practical schemes based on three-move identification protocols were presented in [16]. The principal idea was to mix an undeniable signature related to the confirmer with the hash of the message and the commitment from the signer. Unlike the original idea of Chaum, the authentication of the signer was not independent any longer but merged into the verification of the undeniable signature by the confirmer.

Michels and Stadler [15] showed that one of Okamoto's schemes suffered from a weakness that the confirmer could forge signatures universally. They suggested to use the so called designated confirmer commitments to construct designated confirmer signature schemes. The resulting DCS schemes could be proved secure in the random oracle model, inheriting this property from the use of the Fiat-Shamir paradigm for constructing signatures. However, as pointed out by Camenisch and Michels in [2], their models (and all previous schemes) were vulnerable to an adaptive signature-transformation attack when several signers shared the same confirmer. In order to prevent the attacks, Camenisch and Michels presented a confirmer signature scenario based on RSA signature and Cramer-Shoup public-key encryption scheme.

Using strong witness hiding proofs of knowledge, Goldwasser and Waisbard [8] presented simple transformations of several specific signature schemes into DCS schemes which could be proved secure without resorting to random oracles and without appealing to generic zero-knowledge proofs. Later, Gentry et al. [7] provided an alternate generic transformation to convert any signature scheme into a DCS scheme without adding random oracles. The key technique used was a signature on a commitment and a separate encryption of the random string used for commitment.

Harn and Yang presented (t, n) threshold undeniable signature schemes [9] for the cases $t = 1$ and $t = n$. However, the latter was successfully attacked by Landau [12]. Lin et al. [13] proposed a solution that works for any $t, 1 \leq t \leq n$. Unfortunately, it was flawed as well if signers were not assumed to be honest.

2 Definitions

2.1 Formal Definition for an SDCS

A society-oriented designated confirmer signature scheme involves several entities listed below.

Signer group: A group that consists of n individuals who share the same responsibility of signing messages.

Signing group: Any subgroup that consists of at least t members of the signer group. Each signature is generated by such a group. No other subgroup of less than t members from the signer group nor anyone outside the signer group can produce any valid signature, and is not referred to as a legitimate signing group.

Signing combiner: Any member from the signing group whose task, on behalf of the signing group, is to

1) choose some random values,
2) compute the commitments,
3) encrypt the random values and,
4) collect all the partial results,

to generate the final signature for the receiver.

Verifier: The signature holder who wants to obtain the validity of a signature.

Confirmer group: A group that is composed of l individuals who are designated as confirmers of a specific message signed.

Confirming group: Any subgroup composed of at least k members of the confirmer group, whose members will work together to prove the validity of a signature on a message. Any other subgroup of less than k members from the confirmer group or anyone outside the confirmer group can not provide the validity proof for a signature, and is not referred to as a legitimate confirming group.

Confirming combiner: Any member from the confirming group whose task, on behalf of the confirming group, is to collect partial witness produced by the confirmers and provide the final validity proof of the alleged signature.

Given a message m to be signed, the signing combiner chooses randomly a value r and encrypts it as c, by employing a threshold encryption scheme and with all l public keys of the confirmer group as the encryption keys. Then, a commitment φ on m and r is calculated. Finally, t numbers of the signing group produce the threshold signature σ on the concatenation $\varphi||c$. The resulting SDCS signature is the tuple $\sigma^* = (\varphi, c, \sigma)$.

In the subsequent confirmation or disavowal procedure, a random value r' is worked out from the ciphertext c by k members of the confirming group selected by the verifier, followed by checking whether c is a valid ciphertext for r'. If c is not a valid ciphertext for r', the confirming group proves that its decryption is carried out in an appropriate way and the resulting r' does not agree with the ciphertext c in the SDCS signature. Otherwise the confirmation or disavowal procedure continues to generate the validity proof or invalidity proof for the alleged signature by means of proving the equality or inequality of two discrete logarithms.

More precisely, a society-oriented designated confirmer signature scheme includes the following algorithms and protocols.

- **System Parameters Generation** $PG(2^\lambda)$: An efficient probabilistic algorithm that on input a security parameter λ, outputs system parameters $SP = (P, G, g, h, n, t, l, k, N, p, q, M, H, u, v)$, as explained in next section.

- **Keys Generation** $SKGen(SP) \wedge CKGen(SP)$: This procedure consists of two efficient probabilistic algorithms $SKGen(SP)$ and $CKGen(SP)$, both with the system parameters SP as the input. The former outputs for all members of the signer group secret shares $SK_{S_1}, ..., SK_{S_n}$ of the signing key, the associated verification values $PK_{S_1}, ..., PK_{S_n}$ and a public verifying key PK_S. The latter outputs the public/private key pairs (PK_{C_1}, SK_{C_1}),..., (PK_{C_l}, SK_{C_l}) for all members of the confirmer group.

- **Signature Generation** $SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})$: An efficient algorithm run by the signing group that on input an arbitrary message m, a random value r, all public keys of the confirmer group $PK_{C_1}, ..., PK_{C_l}$, and at least t signing key shares $SK_{S_1}, ..., SK_{S_t}$ of the signing group, outputs the SDCS signature on the message: $\sigma^* = (\varphi, c, \sigma)$.

- **Signature Confirmation or Disavowal** $SV(m, \sigma^*, PK_S, SK_{C_1}, ..., SK_{C_k})$: An efficient protocol between the confirming group and the verifier that on input a message m, a purported SDCS signature σ^*, a public verifying key related to the signer group, at least k private keys $SK_{C_1}, ..., SK_{C_k}$ of the confirming group, outputs the validity or invalidity of the alleged SDCS signature: either 1(true) or 0(false).

2.2 Security Requirements

We use the similar notations in [2] to define the security properties of society-oriented designated confirmer signature schemes.

In the definitions followed, two oracles O_S and O_V will be used. O_S is an oracle which on input m returns a signature σ^* generated according to the probability distribution of $SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})$. O_V is an oracle which on input a message-signature pair (m, σ^*) outputs whether or not (m, σ^*) is correct with respect to the signing secrets of the signing group and the public keys of the confirming group. A message-signature pair (m, σ^*) is correct with respect to the signing secrets of the signing group and the public keys of the confirming group if and only if $Pr[SG(m; r; PK_{C_1}, ..., PK_{C_l};\ SK_{S_1}, ..., SK_{S_t}) = \sigma^*] > 0$, which implies that σ^* could have been generated by $SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})$. Furthermore, M_S and M_V denote the sets of messages that are sent to the oracles O_S and O_V, respectively, during an experiment.

Now, we proceed to the security requirements of SDCS schemes.

- **Completeness of confirmation/disavowal**: If the confirming group and the verifier are honest, then for all $(SK_{S_i}, PK_{S_i}, PK_S) \in \{SKGen(SP)\}(i = 1, ..., n)$, all $(SK_{C_j}, PK_{C_j}) \in \{CKGen(SP)\}(j = 1, ..., l)$, all $m \in \{0,1\}^*$, and all $\sigma^* \in \{0,1\}^*$, it is required that

$$SV(m; \sigma^*; PK_S; SK_{C_1}, ..., SK_{C_k}) = 1,$$

if $\sigma^* \in \{SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})\}$, and

$$SV(m; \sigma^*; PK_S; SK_{C_1}, ..., SK_{C_k}) = 0,$$

if $\sigma^* \notin \{SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})\}$.

- **Soundness of confirmation/disavowal**: For all sufficiently large λ, $(SK_{S_i}, PK_{S_i}, PK_S) \in \{SKGen(SP)\}(i = 1, ..., n)$, all $(SK_{C_j}, PK_{C_j}) \in \{CKGen\ (SP)\}(j = 1, ..., l)$, all $m \in \{0,1\}^*$, and all $\sigma^* \in \{0,1\}^*$, and for every polynomial $p(\cdot)$, it is required that

$$Pr[SV(m; \sigma^*; PK_S; SK_{C_1}, ..., SK_{C_k}) = 0] < 1/p(\lambda),$$

if $\sigma^* \in \{SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})\}$, and

$$Pr[SV(m; \sigma^*; PK_S; SK_{C_1}, ..., SK_{C_k}) = 1] < 1/p(\lambda),$$

if $\sigma^* \notin \{SG(m; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})\}$.

- **Security for signers**: Let A be a polynomial time forging algorithm, on input public keys $PK_{S_1}, ..., PK_{S_n}$ of the signer group, public keys $PK_{C_1}, ..., PK_{C_l}$ of the confirmer group, at most $t-1$ secret keys $SK_{S_1}, ..., SK_{S_{t-1}}$ of the signer group,

and given access to the oracle O_S for polynomially many adaptively chosen inputs of its choice, outputs the SDCS signature $\sigma^* = (\varphi, c, \sigma)$ on the message $m \notin M_S$ of its choice. We require that $Pr[SV(m; \sigma^*; PK_S; SK_{C_1}, ..., SK_{C_k}) = 1] < 1/p(\lambda)$ for all sufficiently large λ.

- **Security for Confirmers**: Let A be a confirming-adversary. Given the public/private key-pairs $(PK_{S_1}, SK_{S_1}), ..., (PK_{S_n}, SK_{S_n})$ of the signer group, public keys $PK_{C_1}, ..., PK_{C_l}$ of the confirmer group, he is allowed to choose any public/private key-pairs $(PK'_{S_1}, SK'_{S_1}), ..., (PK'_{S_n}, SK'_{S_n})$ and to make polynomially many adaptively queries to the oracle O_V about the validity of the message-signature pair $(m_i, \overline{\sigma_i^*} = SG(m_i; r; PK_{C_1}, ..., PK_{C_l}; SK'_{S_1}, ..., SK'_{S_t}))$. Then he may present two messages $m_0, m_1 \notin M_V$ and is given the corresponding $\sigma^* = SG(m_b; r; PK_{C_1}, ..., PK_{C_l}; SK_{S_1}, ..., SK_{S_t})$ for m_b $(b \in \{0, 1\})$. Now the adversary is again allowed to query the oracles O_S and O_V except that σ^* are not allowed in any of these queries. Finally, the adversary must output the value of b. The probability that the adversary gets the right value of b is smaller than $1/2 + 1/p(\lambda)$.
- **Non-transferability of verification**: The confirmation or disavowal should not be transferable. In other words, although he knows whether a given SDCS signature is valid or not by participating in an execution of the interactive verification, the verifier does not obtain any information that could be used to convince a third party about the validity of this SDCS signature.

3 Preliminaries

3.1 Notations

In this subsection, we will list all notations for the parameters involved in our schemes.

P: A large prime $P(2^{159} < P < 2^{160})$.

G: A cyclic group with order of P.

g, h: Random generators of G, with logarithms $log_h g$ and $log_g h$ unknown.

n: The number of members in the signer group.

t: The minimal number of members in a signing group.

l: The number of members in the confirmer group.

k: The minimal number of members in a confirming group.

p, q: Two secret large secure primes s.t. $2^{511} < (p = 2p' + 1) < 2^{512}$ and $2^{511} < (q = 2q' + 1) < 2^{512}$ with p' and q' also primes.

N: An RSA modulus s. t. $N = pq$.

M: The order of the cyclic group QR_N of quadratic residues as $M = p'q'$.

H: A collision-free hash function $\{0,1\}^* \rightarrow \{0,1\}^{1024}$.

H: A collision resistant hash function $\{0,1\}^* \rightarrow Z_P^*$.

e, d: $n < e < min(p', q')$ is the public prime exponent for the signer group while d is the private signing key s.t. $de = 1 \bmod M$.

v: A random generator in the quadratic residue group QR_N.

u: An element in Z_N^* whose Jacobi symbol with respect to N is -1, i.e., $J(\frac{u}{N}) = -1$.

$f(x)$: A random polynomial with degree of $t-1$ as $f(x) = \sum_{i=0}^{t-1} a_i x^i$, where $a_0 = d$ and $a_i \in_R Z_M^*$(i=1,2,...,t-1).

$d_i (i = 1, 2, ..., n)$: The signing-key share evaluated as $d_i = f(i)(n!)^{-1} \bmod M$ which is sent in private to the *i-th* member of the signer group.

$v_i (i = 1, 2, ..., n)$: the public verification value associated with d_i s.t. $v_i = v^{d_i} \bmod N$.

$(SK_{C_i}, PK_{C_i}) (i = 1, 2, ..., l)$: Private/public key pair of the *i-th* confirmer, where $SK_{C_i} \in_R Z_P^*$ and$PK_{C_i} = g^{SK_{C_i}}$.

We sketch a threshold encryption scheme and a threshold signature scheme before we outline the details of the proposed SDCS scheme.

3.2 A (k, l) Threshold Encryption Scheme

In the (k, l) secret sharing schemes, the recovery of the secret depends on the knowledge of k points $(x_1, y_1), ..., (x_k, y_k)$. If we publish only one coordinate of the point (x_i, y_i) and associate the other one with a private key, then we can obtain a (k, l) threshold encryption scheme. In the following scheme, which is a simplified version of that scheme in [10], the public part is the second component of $(PK_{C_i}^r, w_i)$. Only with the private key SK_{C_i} can the first component be worked out as $PK_{C_i}^r = R^{SK_{C_i}} = (g^r)^{SK_{C_i}}$.

For the message $r \in_R Z_P^*$ to be encrypted, the signing combiner:

1) Selects randomly $k-1$ numbers $b_1, ...b_{k-1} \in_R Z_P^*$, and,
2) constructs a polynomial with degree of $k-1$: $F(x) = \sum_{i=0}^{k-1} b_i x^i \in Z_P[x]$, where $b_0 = r$.
3) With l public keys $PK_{C_1}, ..., PK_{C_l}$ of the designated confirmer group, computes the values $R = g^r$ and $w_1 = F(PK_{C_1}^r), ..., w_l = F(PK_{C_l}^r)$.

The ciphertect for the message r is $c = (R, w_1 ..., w_l)$.

To decrypt the ciphertext, at least k out of the l designated confirmers come together for this work.

1) Each confirmer C_j calculates a partial result $c_j = R^{SK_{C_j}} = PK_{C_j}^r$ with his private key SK_{C_j}, and then sends it to the confirming combiner in a secure channel.

2) Upon receiving at least k partial results, the confirming combiner produces the final result, i.e., the message $r = \sum_{j=1}^{k} w_j \prod_{t\in\{1,...k\},t\neq j} \frac{c_t}{c_t - c_j}$.

Notice in the description of the scheme that, any element of G in the form of g^k involved in the evaluations of a polynomial will always be viewed as an integer in Z_P^*.

Note that, without the private keys of the confirmers, no $c_j = R^{SK_{C_j}}$ can be calculated. Moreover, no one can obtain the message r unless at least k pairs of (c_j, w_j) are available.

3.3 A (t, n) Threshold Signature Scheme

Unlike the other kinds of signatures, in applications of the society-oriented designated confirmer signatures, the confirmer groups rather than the signer group vary from signature to signature. In other words, there might exist only one signer group but many confirmer groups in the settings of SDCS. It is natural to set a fixed pair of private signing key and public verifying key for the signer group. But it is not a good idea, from viewpoints of security and cost, to treat the confirmer group in similar manner since it will lead to the distribution of so many keys. Random values are introduced to take the place of fixed keys. Thus the RSA threshold signature scheme presented by Wang e*t al.* in [19], improved from that of Shoup's scheme [18], is applicable for our situations. We apply the improved scheme in the following manner.

It is assumed that all parameters involved have been well generated as listed in Subsection 3.1.

To sign a message m_0 (the concatenation of φ and c in the coming scheme) whose digest is calculated as $m' = H(m_0)u^{(1-J(\frac{H(m_0)}{N}))/2}$,

1) each member of the signer group can compute his partial signature as $\sigma_i = m'^{2d_i} \bmod N$ and provide zero-knowledge proof $Proof_{\sigma_i} = PK\{(\beta) : v_i = v^\beta \wedge \sigma_i = (m'^2)^\beta\}$. The partial signature and the proof are sent securely to the signing combiner.

2) After collecting at least t valid partial signatures $\sigma_1, \sigma_2, ..., \sigma_t$ from the signing group, the signing combiner computes the threshold signature as

$$\sigma = \prod_{i=1}^{t} \sigma_i^{2n! \prod_{j\in\{1,...,t\},j\neq i} \frac{j}{j-i}} \bmod N.$$

A verifier who holds the message-signature pair (m_0, σ) can simply check the validity if the equation $\sigma^e = m'^4 \bmod N$ holds.

4 The Proposed SDCS Scheme

4.1 The Scheme

With the techniques of threshold encryptions and threshold signatures as described above, we construct our society-orient designated confirmer signature scheme as follows.

- **System Parameters Generation** $PG(2^\lambda)$: Given the security parameter λ, produce the system parameters $SP = (P, G, g, h, n, t, l, k, N, M, H, u, v;\ p, q)$. Here, G is a cyclic group with order of $P(2^{159} < P < 2^{160})$. g, h are two random generators of G with logarithms $log_h g$ and $log_g h$ unknown. The number of members in the signer group is set as n, at least t out of which are required to sign a message. The number of members in the confirmer group is set as l, at least k out of which are required to confirm the validity of a signature. The RSA modulus N is the product of two secret large secure primes p, q: $2^{511} < (p = 2p' + 1) < 2^{512}$ and $2^{511} < (q = 2q' + 1) < 2^{512}$ with p' and q' also primes. M is a product of p' and q'. H denotes a collision-free hash function $\{0,1\}^* \rightarrow \{0,1\}^{1024}$. v is a random generator in the quadratic residue group QR_N and u is an element in Z_N^* whose Jacobi symbol with respect to N is -1.

- **Keys Generations** $SKGen(SP) \wedge CKGen(SP)$: Given the system parameters SP, generate the keys related to all players in the scheme. $n < e < min(p', q')$ is the public prime exponent for the signer group while d is the private signing key. $d_i = f(i)(n!)^{-1} \bmod M (i = 1, 2, ..., n)$ is the signing-key share for the *i-th* signer, where $f(x)$ is a random polynomial with degree of $t-1$ as $f(x) = \sum_{i=0}^{t-1} a_i x^i \in Z_P[x]$ and $a_0 = d$. $v_i = v^{d_i} \bmod N (i = 1, 2, ..., n)$ is the public verification value associated with d_i. $(SK_{C_i}, PK_{C_i})(i = 1, 2, ..., l)$ is the private/public key pair of the *i-th* confirmer, where $SK_{C_i} \in_R Z_P^*$ and $PK_{C_i} = g^{SK_{C_i}}$.

- **SDCS Signature Generation**: Let the message to be signed be $m \in Z_P^*$.

  - The signing combiner chooses a random value $r \in_R Z_P^*$, and computes a Pedersen commitment [17] $\varphi = g^m h^r$ on the message m.
  - He calculates the ciphertext c of the random number r under the public keys $PK_{C_1}, ..., PK_{C_l}$ of l members of the confirmer group as

    $$c = En_{PK_{C_1},...,PK_{C_l}}(r) = (R, w_1, ..., w_l),$$

    where $R = g^r$ and $w_j = F(PK_{C_j}^r)$. The random function $F(x)$ is constructed in the same way as described in Section 3.2.

– He sends to each member of the signing group secretly (m, φ, c) with a proof $PK\{(\alpha) : \varphi = g^m h^\alpha \wedge R = g^\alpha \wedge w_1 = PK^\alpha_{C_1} \wedge ... \wedge w_l = PK^\alpha_{C_l}\}$.

– After validating the proof of (m, φ, c), members of the signing group compute $m' = H(\varphi||c)u^{(1-J(\frac{H(\varphi||c)}{N}))/2}$ and the partial signatures $\sigma_i = m'^{2d_i} \bmod N$. Then they sends them secretly back to the signing combiner together with the proofs $PK\{(\beta) : v_i = v^\beta \wedge \sigma_i = (m'^2)^\beta\}$.

– After collecting at least t valid partial signatures $\sigma_1, \sigma_2, ..., \sigma_t$, the signing combiner computes $\sigma = \prod_{i=1}^t \sigma_i^{2n! \prod_{j\in\{1,...,t\}, j\neq i} \frac{j}{j-i}} \bmod N$ and publishes the SDCS signature on the message m as $\sigma^* = (\varphi, c, \sigma)$.

- **SDCS Signature Confirmation or Disavowal**:

1) If σ is a valid signature on m', *i.e.*, $\sigma^e = m'^4 \bmod N$ holds, the verifier submits the message-signature pair $(m, \sigma^* = (\varphi, c, \sigma))$ to k members of the confirming group that he trusts.

2) Each member of the confirming group, with the private key SK_{C_j}, calculates $c_j = R^{SK_{C_j}}$ and sends it, with the proofs $PK_{C_j}\{(\gamma) : c_j = R^\gamma \wedge PK_{C_j} = g^\gamma\}$, secretly to the confirming combiner who will compute a value

$$r' = \sum_{j=1}^{k} w_j \prod_{t\in\{1,...k\}, t\neq j} \frac{c_t}{c_t - c_j}.$$

3) If $R = g^{r'}$ does not hold, $c_j (j = 1, .., t)$ and proofs $PK_{C_j}\{(\gamma) : c_j = R^\gamma \wedge PK_{C_j} = g^\gamma\}$ are sent to the verifier who will decrypt the ciphertext himself and be convinced that he has submit an invalid ciphertext, and the procedure terminates. Otherwise, a bi-proof analogous to that in [14] is executed to prove the equality $log_g R = log_h \frac{\varphi}{g^m}$ or the inequality $log_g R \neq log_h \frac{\varphi}{g^m}$.

4) The verifier is required to choose two random values $\mu, \nu \in Z_P^*$ and compute a commitment value $\alpha = g^\mu R^\nu$. The value α is sent to the confirming combiner.

5) The confirming combiner selects three random values $\kappa, \widetilde{\kappa}, \omega \in Z_P^*$. He then calculates four values $r_g = g^\kappa, \widetilde{r_g} = g^{\widetilde{\kappa}}, r_h = h^\kappa, \widetilde{r_h} = h^{\widetilde{\kappa}}$ and sends them with the value ω to the verifier.

6) The verifier opens for the confirming combiner the values μ, ν committed in the commitment value α.

7) The confirming combiner checks if $\alpha = g^\mu R^\nu$ holds. If does, he computes $s = \kappa - (\nu + \omega)r' \bmod P$, $\widetilde{s} = \widetilde{\kappa} - (\nu + \omega)\kappa \bmod P$ and sends them to the verifier.

8) The verifier first checks whether $g^s R^{\nu+\omega} = r_g$, $g^{\widetilde{s}} r_g^{\nu+\omega} = \widetilde{r_g}$ and $h^{\widetilde{s}} r_h^{\nu+\omega} = \widetilde{r_h}$. If all these equation holds, he is convinced about the validity of the SDCS signature in the following way. If $h^s(\frac{\varphi}{g^m})^{\nu+\omega} = r_h$, then $log_g R = log_h \frac{\varphi}{g^m}$ and the signature $\sigma^* = (\varphi, c, \sigma)$ is a valid SDCS signature on the message m. If $h^s(\frac{\varphi}{g^m})^{\nu+\omega} \neq r_h$, then $log_g R \neq log_h \frac{\varphi}{g^m}$ and the signature $\sigma^* = (\varphi, c, \sigma)$ is not a valid SDCS signature on the message m.

- **Completeness of Confirmation or Disavowal**: Firstly, if σ in the SDCS signatue σ^* is a valid signature on the message $m' = H(\varphi||c)u^{(1-J(\frac{H(\varphi||c)}{N}))/2}$, we have

$$\begin{aligned}
\sigma^e &= (\prod_{i=1}^{t} \sigma_i^{2n! \prod_{j\in\{1,...,t\}, j\neq i} \frac{j}{j-i}} \bmod N)^e \\
&= (\prod_{i=1}^{t} (m'^{2d_i})^{2n! \prod_{j\in\{1,...,t\}, j\neq i} \frac{j}{j-i}} \bmod N)^e \\
&= (\prod_{i=1}^{t} m'^{4d_i n! \prod_{j\in\{1,...,t\}, j\neq i} \frac{j}{j-i}} \bmod N)^e \\
&= (m'^{4\sum_{i=1}^{t} d_i n! \prod_{j\in\{1,...,t\}, j\neq i} \frac{j}{j-i}} \bmod N)^e \\
&= (m'^{4d} \bmod N)^e \\
&= m'^4 \bmod N.
\end{aligned}$$

Secondly, if c is a ciphertext of the random value r encrypted under the public keys of the confirmer group with $R = g^r$ and $w_i = F(PK^r_{C_i})$ $(i = 1, ..., l)$, then k out of l points can be calculated by k confirmers with their secret keys $(R^{SK_{C_j}}, w_j) = (PK^r_{C_i}, w_j)$. With these points, the coefficient $r = b_0$ of the function can be recovered in this way:

$$\begin{aligned}
r &= \sum_{j=1}^{k} w_j \prod_{t\in\{1,...k\}, t\neq j} \frac{R^{SK_{C_t}}}{R^{SK_{C_t}} - R^{SK_{C_i}}} \\
&= \sum_{j=1}^{k} w_j \prod_{t\in\{1,...k\}, t\neq j} \frac{PK^r_{C_t}}{PK^r_{C_t} - PK^r_{C_i}}.
\end{aligned}$$

Finally, with the knowledge of r, it is not hard to convince the verifier if $\varphi = g^m h^r$ holds with the bi-proof protocol. The protocol is proved in [14] to be complete, sound, and zero-knowledge under the assumption that there is no algorithm running in expected polynomial time to decide whether two discrete logarithms are equal, with non-negligible probability better than guessing.

4.2 Convertibility

If the signature is needed to be selectively turned into a universally verifiable one, the confirming group can produce the signature proof as follows. Two random values $\kappa, \widetilde{\kappa} \in Z_P^*$ are chosen and four values $r_g = g^\kappa, \widetilde{r_g} =$

$g^{\widetilde{\kappa}}, r_h = h^{\kappa}, \widetilde{r_h} = h^{\widetilde{\kappa}}$ are computed by a confirming group. Then it sets $\nu = \mathsf{H}(g, R, h, \frac{\varphi}{g^m}, r_g, r_h, \widetilde{r_g}, \widetilde{r_h})$ and computes $s = \kappa - \nu r'$ and $\widetilde{s} = \widetilde{\kappa} - \nu k$. The signature proof appears as $\sigma = (s, \widetilde{s}, \nu)$. The signature holder can verify the signature by checking the equality

$$\nu = \mathsf{H}(g, R, h, \frac{\varphi}{g^m}, g^s R^{\nu}, h^s (\frac{\varphi}{g^m})^{\nu}, g^{s+\widetilde{s}} R^{\nu}, h^{s+\widetilde{s}} (\frac{\varphi}{g^m})^{\nu})$$

On the other hand, if the non-interactive zero-knowledge proof can only be provided to a designated verifier, the public key $PK_V = g^{SK_V}$ of the verifier is necessary to construct such a designated verifier proof in a similar manner as the preceding construction of the convertible signature proof. The main difference is that two more random values α, β are chosen to determine a trapdoor commitment $C = PK_V^{\alpha} h^{\beta}$ which will be set as an additional input of the hash function H.

5 Security Analysis

In the SDCS scheme, given a message-signature pair $(m, \sigma^* = (\varphi, c, \sigma))$, anyone can verify whether the ordinary signature σ is a valid signature on the combination of φ and c. However, only at least k confirmers can work together to check if φ is a right commitment on the message m and the plaintext associated to the ciphertext c. In other words, only the confirming group can obtain the complete relation between the signature and the message.

With the technology of threshold signatures, the signature σ can be generated by a group with at least t signers. Given the public keys of the signer group and those of the confirmer group, nobody outside the signer group, nor any group of no more than $t-1$ signers in collusion, even with the secret keys of the confirmer group, can produce a valid threshold signature. Moreover, partial signatures $\sigma_i = m'^{2d_i}$ rather than secret shares of signing key are sent to the signing combiner, in the signing procedure. No secret shares can be obtained from the partial signatures, so the signing combiner can not recover the signing key on his own and has no advantage over the other signing members in this procedure. This prevents the signing combiner from signing any message by himself next time. In such a way, unforgeability of signatures is guaranteed.

The relation between the commitment φ and the original message m depends on a random value r, i.e., the plaintext of the ciphertext c which is produced by a threshold encryption scheme. The fact that c is a ciphertext for the plaintext r or on the contrary can only be confirmed by the decrypting group, i.e., the confirming group composed of at least k confirmers. Nobody outside the confirmer group, nor any group of no more than $k-1$ confirmers in collusion, can do the same thing. Moreover, what are sent to the confirming combiner are the partial decryption results $c_j = R^{SK_{C_j}}$, from which no secret information is released. So, the confirming combiner can not decrypt any other ciphertexts on his own. The random value recovered this time does not help the next confirmation since another random value will be used. This means that the confirming combiner has no advantage over the other confirming members. Unlike the signing procedure, no fixed decrypting keys are shared by the confirmer group, which eliminates the cost of distribution and storage of secret shares. There exists no thread against the exposure of the secret decrypting keys. In such a way, the restrictive confirmation is achieved.

Upon obtaining the random value by decrypting the ciphertext, the confirming group proves the validity or invalidity of the alleged SDCS signature to the verifier in a zero-knowledge way. The transcript of the proof of knowledge will not convince a third party that σ^* is an SDCS signature on the message m or on the contrary. This is because the verifier can produce, at his will, a similar transcript to validate or invalidate an alleged SDCS signature in the following manner. He firstly chooses randomly $\mu, \nu, \omega, s, \widetilde{s} \in Z_P^*$ and computes $\alpha = g^{\mu} R^{\nu}$, $r_g = g^s R^{\nu+\omega}$, $\widetilde{r_g} = g^{\widetilde{s}} r_g^{\nu+\omega}$. Then, if he wants to validate the signature, he calculates $r_h = h^s (\frac{\varphi}{g^m})^{\nu+\omega}$ and $\widetilde{r_h} = h^{\widetilde{s}} r_h^{\nu+\omega}$. And if he wants to invalidate the signature, he selects randomly $r_h \neq h^s (\frac{\varphi}{g^m})^{\nu+\omega}$ and calculates $\widetilde{r_h} = h^{\widetilde{s}} r_h^{\nu+\omega}$. Therefore, nobody will believe of the transcript $(\mu, \nu, \omega, s, \widetilde{s}, r_g, \widetilde{r_g}, r_h, \widetilde{r_h})$ the verifier provides and we have the property of non-transferability of verification.

Of course, as mentioned in Section 4, the proofs can be changed into non-interactive ones that everyone or only a designated individual can verify the signature. In such a way, the SDCS signature becomes an ordinary or a designated verifier (threshold) signature.

6 Conclusions

The designated confirmer signature can only be verified with the help of the confirmer designated by the signer, which releases the signer from the task of verification. We consider the applications of designated confirmer signatures in the society-oriented situations, where the signature is produced by a group of signers and is verifiable only with the help of a group of designated confirmers. The model and the security requirements of the society-oriented designated confirmer signature are proposed. A concrete realization is provided with appropriate security analysis. Ordinary designated confirmer signatures and threshold designated confirmer signature schemes, where the messages are signed by one individual, can be viewed as special instances of our SDCS schemes.

Acknowledgments

We are grateful to the anonymous referees for their invaluable suggestions to improve this paper.

This work was supported by the National Natural Science Foundation of China (No.60773202,60572059,60503006,90604009), NSFC-KOSEF Joint Research Project (No.60611140543) and 863 Program (No.2006AA01Z267).

References

[1] J. Boyar, D. Chaum, I. B. Damgard, and T.P. Pedersen, "Convertible undeniable signatures," *Cryptology, Crypto '90*, pp. 189-205,1990.

[2] J. Camenisch, and M. Michels, "Confirmer signature schemes secure against adaptive adversaries," *Cryptology, Eurocrypt '00*, pp. 243-258, 2000.

[3] D. Chaum, "Designated confirmer signatures," *Cryptology, Eurocrypt '94,* pp. 86-91, 1994.

[4] D. Chaum, and H. V. Antwerpen, "Undeniable signatures," *Cryptology, Crypto '89*, pp. 212-216, 1989.

[5] D. Chaum, E. v. Heijst, and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer," *Cryptology, Crypto '91,* pp. 470-484, 1991.

[6] W. Diffie, and M. E. Hellman, "New direction in cryptography," *IEEE Transactions on Information Theory,* vol. IT-22, pp. 644-654,1976.

[7] C. Gentry, D. Molnar, and Z. Ramzan1, "Efficient designated confirmer signatures without random oracles or General Zero-Knowledge Proofs," *Cryptology, Asiacrypt 2005,* pp. 662-681, 2005.

[8] S. Goldwasser, and E. Waisbard, "Transformation of digital signature schemes into designated confirmer signature schemes," *Theory of Cryptography Conference (TCC 2004),* pp. 77-100, 2004.

[9] L. Harn, and S. Yang, "Group-oriented undeniable signature schemes without the assistance of a mutually trusted party," *Cryptology, Auscrypt '92,* pp. 133-142, 1993.

[10] T. Hwang, "Cryptosystem for group oriented cryptography," *Cryptology, Eurocrypt '90,* pp. 352-360, 1990.

[11] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," *Cryptology, Eurocrypt '96,* pp. 143-154, 1996.

[12] S. Landau, "Weakness in some threshold cryptosystems," *Cryptology, Crypto '96,* pp. 74-83, 1996.

[13] C. H. Lin, C. T. Wang, and C. C. Chang, "A group-oriented (t, n) undeniable signature scheme without trusted center," *Australasian Conference on Information Security and Privacy (ACISP '96),* pp. 266-274, 1996.

[14] M. Michels, and M. Stadler, "Efficient convertible undeniable signature schemes," *Selected Areas in Cryptography (SAC '97),* pp. 231-243, 1997.

[15] M. Michels, and M. Stadler, "Generic constructions for secure and efficient confirmer signature schemes," *Cryptology, Eurocrypt '98,* pp. 458-464, 1998.

[16] T. Okamoto, "Designated confirmer signatures and public-key encryption are equivalent," *Cryptology, Crypto '94,* pp. 61-74, 1994.

[17] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Cryptology, Crypto '91,* pp. 129-140, 1992.

[18] V. Shoup, "Practical threshold signatures," *Cryptology, Eurocrypt '00,* pp. 207-220, 2000.

[19] G. Wang, S. Qing, M. Wang, and Z. Zhou, "Thrshold undeniable RSA signature scheme," *International Conference on Information and Communications Security (ICICS 2001)*, pp. 221-232, 2001.

Baodian Wei is an Associate Professor in the Department of Electronics and Communication Engineering at Sun Yan-sen University in Guangzhou, China. He obtained his Ph. D degree from School of Information and Communication Engineering, Xidian University in 2004. His research interests include cryptology and its applications.

Fangguo Zhang is a Professor in the Department of Electronics and Communication Engineering at Sun Yan-sen University in Guangzhou, China. He obtained his Ph. D. degree in Cryptography from School of Communication Engineering, Xidian University in 2001. His main research interests include elliptic curve cryptography, pairing-based cryptosystem and its applications.

Xiaofeng Chen is an Associate Professor in the Department of Computer Science at Sun Yan-sen University, Guangzhou, China. He obtained his Ph. D. degree in Cryptography from School of Communication Engineering, Xidian University in 2003. His main research interests include public key cryptography and E-commerce security.