Efficient Designated Confirmer Signature from Bilinear Pairings

Fangguo Zhang
School of Information Science and Technology
Sun Yat-Sen University,
Guangzhou 510275,
P.R.China
Guangdong Key Laboratory of Information Security Technology
isszhfg@mail.sysu.edu.cn

Xiaofeng Chen
School of Information Science and Technology
Sun Yat-Sen University,
Guangzhou 510275,
P.R.China
Guangdong Key Laboratory of Information Security Technology
isschxf@mail.sysu.edu.cn

Baodian Wei
School of Information Science and Technology
Sun Yat-Sen University,
Guangzhou 510275,
P.R.China
Guangdong Key Laboratory of Information Security Technology
weibd@mail.sysu.edu.cn

ABSTRACT

Designated confirmer signature is an important cryptographic primitive, it is widely used in E-commerce. In this paper, we propose a new designated confirmer signature scheme which is transformed from a new signature scheme. The proposed scheme has very simple construction and is much more efficient than the previous ones and does not need any commitment scheme or strong witness hiding proofs.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption – Public key cryptosystems

General Terms

Design, Theory

Keywords

Signature, Designated confirmer signature, Bilinear pairings

1. INTRODUCTION

Chaum and van Antwerpen [8] introduced the notion of undeniable signatures which allows to authenticate a message in such a way that the recipient has to interact with the signer in order to be convinced of its validity. The undeniable signatures rely on the signer to be available and not to act maliciously, otherwise, the recipient cannot make use of the signature. Chaum [9] introduced the notion of a designated confirmer signatures (DCS), which overcome the limitations of undeniable signatures. DCS require the assistance of a trusted third party called the confirmer. Given a signature σ that the signer issues, the confirmer can execute a special Confirm protocol to prove that a signature σ is a valid signature on a message m, or a Deny protocol to show that σ is not a valid signature on a message m. Without the confirmer, however, no party can determine whether σ is a valid signature for m or not.

Since the invention of DCS, several concrete realizations of designated confirmer signatures were presented [6, 11, 12, 13, 14, 15]. Okamoto provided [15] the first formal definition for a DCS scheme and showed constructively that a DCS scheme is equivalent to a public-key encryption scheme with respect to existence. Using strong witness hiding proofs of knowledge, Goldwasser and Waisbard [12] presented simple transformations of several specific signature schemes into DCS schemes. Later, Gentry et al. [11] provided an alternate generic transformation to convert any signature scheme into a DCS scheme without adding random oracles. The key techniques used are a signature on a commitment and a separate encryption of the random string used for commitment. At PKC 2007, Wang et al. [17] showed that Gentry et al.'s DCS transformation does not meet the desired security requirements by identifying two security flaws and pointed out the reasons that cause those flaws and proposed a secure improvement to fix the flaws. Wei [18] also points out the security flaws of [11] and [12] in his paper showing that their scheme does not meet essential security requirements under their security model. He shows successful signature transformation attacks on these schemes, and gives amendments to improve these schemes into invisible DCS.

In recent years, the bilinear pairings have been found to be very useful in various applications in cryptography and have allowed us to construct new cryptographic primitives. Especially, the bilinear pairings can be used to construct short signature scheme, including secure signature scheme without random oracle. Boneh, Lynn and Shacham [4] proposed the first provably secure short signature scheme (BLS scheme) in the random oracle model provided that the computational Diffie-Hellman problem (CDHP) is intractable. A very smart application of BLS scheme is to construct the universal designated-verifier signatures (UDVS) [16]. Another two short signature schemes, known as ZSS [22] and BB04 [2] schemes, depend on the stronger Diffie-Hellman assumption. The most impressive application of them is to construct the verifiably encrypted signature (VES) [21].

At VietCrypt 2006, Zhang et al. proposed a new signa-

ASIACCS '08, March 18-20, Tokyo, Japan

ture scheme from bilinear pairings, which we called ZCSM scheme [20]. In this paper, we will show that we can easily construct an efficient DCS scheme using ZCSM signature scheme. Compared with the previous generic transformation method, our approach does not need any commitment scheme or strong witness hiding proofs.

2. PRELIMINARIES

2.1 Bilinear Pairings

We briefly review the bilinear pairings using the same notation as in [3, 4]:

Let $\mathbb{G}$ be a (mutiplicative) cyclic group of prime order q. Let g be a generator of $\mathbb{G}$.

Definition: A map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ (here $\mathbb{G}_T$ is another mutiplicative cyclic group such that $|\mathbb{G}| = |\mathbb{G}_T| = q$) is called a bilinear pairing if this map satisfies the following properties:

1. **Bilinearity:** For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, we have $e(u^a,\ v^b) = e(u,\ v)^{ab}$.
2. **Non-degeneracy:** $e(g, g) \neq 1$. In other words, if g is a generator of $\mathbb{G}$, then $e(g,\ g)$ generates $\mathbb{G}_T$.
3. **Computability:** There is an efficient algorithm to compute $e(u,\ v)$ for all $u, v \in \mathbb{G}$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ is efficiently computable and there exists a group $\mathbb{G}_T$, and a bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ as above.

2.2 Proofs of Knowledge of Discrete Logarithms

We will use the notation introduced by Camenisch and Stadler [5] for various proofs of knowledge of discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \gamma) : y = g^{\alpha} h^{\beta} \wedge z = g'^{\alpha} h'^{\gamma}\};$$

is used for proving the knowledge of integers α, β and γ such that $y = g^{\alpha} h^{\beta}$ and $z = g'^{\alpha} h'^{\gamma}$ holds. Here y, g, h, z, g' and h' are elements of some groups $\mathbb{G} =< g >=< h >$ and $\mathbb{G}_T =< g' >=< h' >$.

2.3 The $k+1$ Square Roots Assumption

We give the definitions of $k+1$ square roots problem and its assumption that firstly introduced in [20] as follows:

Definition 1. The k+1 **Square Roots Problem** in $(\mathbb{G}, \mathbb{G}_T)$ is as follows: For an integer k, and $x \in_R \mathbb{Z}_q$, $g \in \mathbb{G}$, given

$$g, \alpha = g^x, h_1, \ldots, h_k \in \mathbb{Z}_q, g^{(x+h_1)^{\frac{1}{2}}}, \ldots, g^{(x+h_k)^{\frac{1}{2}}},$$

compute $g^{(x+h)^{\frac{1}{2}}}$ for some $h \notin \{h_1, \ldots, h_k\}$.

Definition 2. We say that the $(k+1, t, \epsilon)$-square roots assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ if no t-time algorithm has advantage at least ϵ in solving the k+1-square roots problem in $(\mathbb{G}, \mathbb{G}_T)$, i.e., $k+1$-square roots problem is (t, ϵ)-hard in $(\mathbb{G}, \mathbb{G}_T)$.

2.4 ZCSM Signature Scheme from Pairings

We describe ZCSM signature scheme as follows:

The system parameters are $(\mathbb{G},\ \mathbb{G}_T,\ e,\ q,\ g)$.

Key Generation. Randomly select $x, y \in_R \mathbb{Z}_q^*$, and compute $u = g^x,\ v = g^y$. The public key is $(u,\ v)$. The secret key is (x, y).

Signing: Given a secret key $x, y \in_R \mathbb{Z}_q^*$, and a message $m \in \mathbb{Z}_q$, pick a random $r \in_R \mathbb{Z}_q^*$, and compute

$$\sigma = g^{(x+my+r)^{\frac{1}{2}}} \in \mathbb{G}.$$

Here $(x+my+r)^{\frac{1}{2}}$ is computed modulo q. When $x+my+r$ is not a quadratic residue modulo q we try again with a different random r. The signature on message m is (σ, r).

Verification: Given a public key $(\mathbb{G},\ \mathbb{G}_T,\ q,\ g,\ u,\ v)$, a message $m \in \mathbb{Z}_q^*$, and a signature (σ, r), verify that

$$e(\sigma,\ \sigma) \stackrel{?}{=} e(uv^m g^r,\ g).$$

For the security, we have

THEOREM 1. *[20] Suppose the $(k+1, t', \epsilon')$-square roots assumption holds in $(\mathbb{G}, \mathbb{G}_T)$. Then the signature scheme above is (t, q_S, ϵ)-secure against existential forgery under an adaptive chosen message attack provided that*

$$q_S < k+1,\ \epsilon = 2\epsilon' + 4\frac{q_S}{q} \approx 2\epsilon',\ t \leq t' - \Theta(q_S T).$$

where T is the maximum time for computing a square root in $\mathbb{Z}_q^*$ and an exponentiation in $\mathbb{G}$.

3. SECURE DESIGNATED CONFIRMER SIGNATURE

We describe designated confirmer signatures following the exposition of [12]. A DCS scheme consists of three participants: a signer **S**, a verifier **V**, and a designated confirmer **C** and is composed of the following algorithms:

- **Setup:** The setup includes two probabilistic polynomial time key generation algorithms: generating two pairs of keys $K^S := (PK_S, SK_S)$ and $K^C := (PK_C, SK_C)$ for **S** and **C**.
- **Sign:** On the input of the signer's secret key SK_S, the (probabilistic) polynomial time algorithm **Sign** generates a signature $\sigma = Sign(m, SK_S)$ for a message m.
- **Verify:** Takes as input (m, σ, PK_S) and outputs Accept if σ is an output of $Sign(m, SK_S)$.
- **ConfirmedSign:** On the input of the signer's secret key SK_S and confirmer's public key PK_C , the (probabilistic) polynomial time algorithm **ConfirmedSign** generates a signature $\sigma' = \text{ConfirmedSign}(m, SK_S, PK_C)$ of m.
- **Confirm:** Let (m, σ') be a supposedly valid message-signature pair. **Confirm** is an interactive protocol between **C** and **V**. At the end of the protocol, it outputs a boolean value which tells whether σ' is accepted as a valid signature of m.
- **Deny:** Let (m, σ') be an alleged invalid message-signature pair. **Deny** is an interactive protocol between **C** and **V**. At the end of the protocol, it outputs a boolean value which tells whether σ' is accepted as an invalid signature.
- **Extract:** takes as input (m, σ', SK_C, PK_S) and returns a string σ such that $Verify(m, \sigma, PK_S)$ outputs Accept if σ is an output of $Sign(m, SK_S)$, and outputs $\perp$ otherwise.

For the security requirements of DCS scheme, we have separate security requirements for signers and confirmers [12]. Similar to the description for the security of DCS in [12], we give a formal definition as follows:

a **Security for signers/Unforgeability:** A DCS is unforgeable against an adaptive chosen message attack for signers if it is infeasible for a forger who only knows the public keys PK_S, PK_C to produce a valid confirmed message-signature pair after executing **ConfirmedSign**, **Confirm** and **Deny** for polynomially many adaptively chosen inputs of its choice.

Formally, for every probabilistic polynomial time forger algorithm $\mathcal{F}$ there exist no non-negligible probability ϵ such that

$$\mathbf{Adv}(\mathcal{F}) =$$

$$\Pr\left[\begin{array}{l} \langle K^S, K^C\rangle \leftarrow \langle \mathbf{Setup}\rangle(1^l); \\ for \text{ i} = 1, 2, \ldots, k, \\ m_i \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_{i-1}, \sigma_{i-1}), \\ \sigma_i \leftarrow \mathbf{ConfirmedSign}(sk, m_i), \\ \mathbf{Confirm}(PK_S, PK_C, m_i, \sigma_i) = accept \\ or\ \mathbf{Deny}; \\ \langle m, \sigma'\rangle \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_k, \sigma_k), \\ m \notin \{m_1, \ldots, m_k\}, \\ \mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept \end{array}\right] \geq \epsilon.$$

Even the forger algorithm $\mathcal{F}$ knows the secret key SK_C of designated confirmer **C**, the probability ϵ is still negligible.

b **Security for designated confirmers:** Assume that $\mathcal{A}$ is a probabilistic polynomial time attacking algorithm which, on input strings $1^l, PK_S, PK_C$ can request the execution of **ConfirmedSign**, **Confirm** and **Deny** for polynomially many inputs of his choice and finally, for a pair (m, σ') of his choice, $\mathcal{A}$ executes $\mathbf{Confirm}(PK_S, PK_C, m, \sigma')$. For all such $\mathcal{A}$, the probability

$$\Pr(\mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept)$$

is negligible.

Even a coalition of Signers, who share the same Confirmer, cannot confirm a DCS that was signed with respect to SK_S.

4. AN EFFICIENT DCS SCHEME

A straightforward way to construct a DCS scheme was suggested by Okamoto [15] using standard cryptographic primitives, such as public-key encryption schemes and digital signature schemes: firstly, the message m 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 of m. Since the signature is encrypted, only the designated confirmer can be convinced of its validity. Moreover, the designated confirmer can easily extract an ordinary signature from it. In order for the recipient to be convinced of the validity of the DCS, the signer and recipient interact in a zero knowledge proof in which the signer proves to the verifier that what he got is indeed an encryption of an ordinary signature of m. Since the last assertion is an NP statement, the general zero-knowledge proofs for NP statements should be used in such constructions.

The above construction is very simple and intuitive and can be easily proved to be secure. As claimed in [12], the problem is that we do not know of efficient zero-knowledge proofs for such an assertion for concrete schemes. Until now, no scheme follows the above construction. In this section, for the first time in the literature, we present a concrete implementation for the above general construction of DCS.

Based on ZCSM signature scheme, a simple and efficient DCS scheme can be constructed as follows:

- **Setup:** This is same as ZCSM signature scheme. The system parameters are $(\mathbb{G},\ \mathbb{G}_T,\ e,\ q,\ g)$. The public key of **S** is $(u,\ v)$, the secret key is (x, y). **C**'s public key is $\alpha = g^{x_c}$, x_c is **C**'s secret key.

- **Sign:** Using ZCSM **Signing**, for a message $m \in \mathbb{Z}_q$, the signature is (σ, r).

- **Verify:** This is same as the **Verification** of ZCSM.

- **ConfirmedSign:** Given the public key $\alpha = g^{x_c} \in \mathbb{G}$ of the designated confirmer **C**, a message $m \in \mathbb{Z}_q$, the signer **S** picks a random $r \in_R \mathbb{Z}_q^*$, and computes

  $$\sigma' = \alpha^{(x+my+r)^{\frac{1}{2}}} \in \mathbb{G}.$$

  Here $(x + my + r)^{\frac{1}{2}}$ is computed modulo q. When $x + my + r$ is not a quadratic residue modulo q we try again with a different random r. The designated confirmer signature on m is (σ', r).

- **Confirm:** **C** first checks that (σ', r) has been signed with **S** using the provided **C**'s public key α, and aborts if the check fails.

  $$e(\sigma',\ \sigma') \overset{?}{=} e(uv^m g^r,\ \alpha)^{x_c}.$$

  Then, **C** performs an interactive zero-knowledge proof with the verifier **V** for knowledge

  $$\log_{e(uv^m g^r,\ \alpha)} e(\sigma',\ \sigma') = \log_{e(g,\ g)} e(\alpha,\ g)$$

  This is an interactive zero-knowledge proof system for the equality of two discrete logarithms [10]. For the details, please refer to [10].

- **Deny:** To disavow a purported signature (σ', r) on m, **C** performs an interactive zero-knowledge proof with the verifier **V** for proving that the discrete logarithms $\log_{e(uv^m g^r,\ \alpha)} e(\sigma',\ \sigma')$ and $\log_{e(g,\ g)} e(\alpha,\ g)$ are unequal. We borrow the method in [7] and provide such a proof as follows:

  1. The confirmer **C** chooses $s \in_R \mathbb{Z}_q^*$, computes the auxiliary commitment

     $$C = (e(uv^m g^r,\ \alpha)^{x_c} / e(\sigma',\ \sigma'))^s,$$

     and sends C to the verifier.

  2. **C** executes the protocol denoted

     $$PK\{(\gamma, \beta) : C = e(uv^m g^r,\ \alpha)^\gamma (\tfrac{1}{e(\sigma',\ \sigma')})^\beta \wedge 1 = e(g,\ g)^\gamma (\tfrac{1}{e(\alpha,\ g)})^\beta\}$$

     with the verifier.

3. The verifier accepts if it accepts in step 2, and if $C \neq 1$; otherwise, the verifier rejects.

- **Extract:** For (m, σ', r), **C** can extract an ordinary ZCSM signature on m using his secret key x_c:

$$\sigma = \sigma'^{x_c^{-1}}.$$

5. ANALYSIS OF THE DCS SCHEME

5.1 Completeness

The completeness can be justified by the following equations:

1. For the verification of **ConfirmedSign** by the confirmer:

$$\begin{aligned} e(\sigma', \ \sigma') &= e(\alpha^{(x+my+r)^{\frac{1}{2}}}, \ \alpha^{(x+my+r)^{\frac{1}{2}}}) \\ &= e(g^{x_c}, \ g^{x_c})^{(x+my+r)^{\frac{1}{2}} \cdot (x+my+r)^{\frac{1}{2}}} \\ &= e(g, \ g^{x_c})^{x_c \cdot (x+my+r)} \\ &= e(g^{x+my+r}, \ \alpha)^{x_c} \\ &= e(uv^m g^r, \ \alpha)^{x_c} \end{aligned}$$

2. For the confirm and deny protocol, if (σ', r) is a valid designated confirmer signature on m, i.e., $e(\sigma', \ \sigma') = e(uv^m g^r, \ \alpha)^{x_c}$, then **C** can give the confirm protocol using any protocol for the equality of two discrete logarithms. Otherwise, **C** can give the deny protocol using any protocol for the inequality of two discrete logarithms. For the correctness of the protocol, please refer to [7].

3. For **Extract**,

$$\sigma'^{x_c^{-1}} = (\alpha^{(x+my+r)^{\frac{1}{2}}})^{x_c^{-1}} = g^{(x+my+r)^{\frac{1}{2}}} = \sigma.$$

   This is an ordinary ZCSM signature on m.

5.2 Security

We show that the proposed scheme satisfies the properties of a designated confirmer signature scheme.

Lemma 1. *If there exists a $(t, q_{CS}, q_C, q_D, \epsilon)$-forger $\mathcal{F}$ using adaptive chosen message attack for the proposed DCS scheme which executing q_{CS} times* **ConfirmedSign***, q_C times* **Confirm** *and q_D times* **Deny***, then there exists a $(t', q_S = q_C, \epsilon)$-forger $\mathcal{F}$ for ZCSM signature scheme. Here $t' = t + T$ and T is the time for computing an exponentiation in $\mathbb{G}$.*

Proof. We consider the strongest case, i.e., the forger algorithm $\mathcal{F}$ knows the secret key SK_C of designated confirmer **C**. Let $(\mathbb{G}, \ \mathbb{G}_T, \ e, \ q, \ g)$ be the system parameters of ZCSM signature, the public key is $(u, \ v)$. We set $PK_S = (u, \ v)$, $PK_C = g^{x_c}$.

Suppose that there exists a forger $\mathcal{F}$, after executing q_{CS} times **ConfirmedSign**, q_C times **Confirm** and q_D times **Deny** for adaptively chosen inputs of its choice, $\mathcal{F}$ can output a forgery (m, σ') on message m with probability at least ϵ in time t, such that

$$\mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept.$$

Here m has never been requested by $\mathcal{F}$ to **ConfirmedSign**.

Let $\sigma = \sigma'^{x_c^{-1}}$, then we have a forgery on ZCSM signature scheme. This is because of

$$e(\sigma, \ \sigma) = e(uv^m g^r, g).$$

□

So, from the Theorem 1 and above Lemma 1, we have

Theorem 2. *The proposed DCS scheme is secure against existential forgery under an adaptive chosen message attack for signers.*

The security of proposed DCS scheme for designated confirmers is obtained from the following theorem:

Theorem 3. *Under the assumption that the discrete logarithm problem is hard in $\mathbb{G}$ and $\mathbb{G}_T$, for any probabilistic polynomial time adversaries $\mathcal{A}$, the probability*

$$\Pr(\mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept)$$

is negligible.

Proof. Suppose that there exicts an adversary $\mathcal{A}$ which on input strings $1^l, PK_S, PK_C$ can request the execution of **ConfirmedSign** for polynomially many inputs of his choice and finally, for a pair (m, σ') of his choice, $\mathcal{A}$ executes $\mathbf{Confirm}(PK_S, PK_C, m, \sigma')$ with the non-negligible probability

$$\Pr(\mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept).$$

Then we will use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ to solve the discrete logarithm problem in $\mathbb{G}$.

Suppose $\mathcal{B}$ is given a challenge:
"Given $g, \ \alpha = g^a \in \mathbb{G}$, compute $a \in \mathbb{Z}_q$"

Now $\mathcal{B}$ sets $(\mathbb{G}, \ \mathbb{G}_T, \ e, \ q, \ g)$ to be the system parameters and set $PK_S = (u, \ v)$, $PK_C = \alpha$. $\mathcal{B}$ can play the role of the signer in **ConfirmedSign**. For any valid confirmed signature (m, σ'), where

$$\sigma' = \alpha^{(x+my+r)^{\frac{1}{2}}} \in \mathbb{G},$$

$\mathcal{B}$ plays the role of the verifier and executes $\mathbf{Confirm}(PK_S, PK_C = \alpha, m, \sigma')$ with $\mathcal{A}$, $\mathcal{A}$ will confirm it using an interactive zero-knowledge proof with a non-negligible probability.

Since **Confirm** is an interactive zero-knowledge proof, i.e., an interactive zero-knowledge proof system for the equality of two discrete logarithms

$$\log_{e(uv^m g^r, \ \alpha)} e(\sigma', \ \sigma') = \log_{e(g, \ g)} e(\alpha, \ g), \ (*)$$

So, using the "Reset Lemma" (or rewrite technique) formulated in [1], it is not hard to find x_c, the discrete logarithm of α. For details, $\mathcal{B}$ selects $c \in \mathbb{Z}_q$ as the challenge and runs the interactive zero-knowledge proof for $(*)$ with $\mathcal{A}$ to obtain its response $t \in \mathbb{Z}_q$. Then $\mathcal{B}$ runs the interactive zero-knowledge proof again with the same state as before but with different challenge $c' \in \mathbb{Z}_q$, obtains its response $t' \in \mathbb{Z}_q$. $\mathcal{B}$ now can extract $x_c = \frac{t-t'}{c'-c} \mod q$.

Therefore, if the discrete logarithm problem is hard in $\mathbb{G}$, then for any probabilistic polynomial time adversaries $\mathcal{A}$, the probability

$$\Pr(\mathbf{Confirm}(PK_S, PK_C, m, \sigma') = accept)$$

is negligible. □

Notice, at the above proof of Theorem 3, we assume that the adversary $\mathcal{A}$ is "passive", i.e., $\mathcal{A}$ did not request the execution of **Confirm** and **Deny** with the simulator. It seems that it is not easy to give a security proof when the adversary $\mathcal{A}$ is adaptive, we remain an interesting problem to find a such proof in the further work.

5.3 Efficiency

The proposed new designated confirmer signature scheme can be implemented by elliptic curve over finite fields. So, for the **ConfirmedSign**, it only needs one elliptic curve point multiplication.

We note that the computation of the pairing is the most time-consuming in pairing based cryptosystems. We can pre-compute $a = e(u,\ g)$, $b = e(g,\ g)$ and $c = e(v,\ g)$, and publish them as part of the signer's public key. At the same time, we can pre-compute $a' = e(u,\ \alpha)$, $b' = e(g,\ \alpha)$ and $c' = e(v,\ \alpha)$, and publish them as part of the designated confirmer's public key. Therefore, in the confirm and deny protocol, **C** and **V** only compute some exponentiations. The confirm protocol in our resulting scheme requires 8 exponentiations (compared to 10 for Gentry et al. and 320 for Goldwasser-Waisbard) and our disavow protocol requires one pairing and at most 20 exponentiations (compared to 41 for Gentry et al. and using a general zero-knowledge proof for Goldwasser-Waisbard).

Our designated confirmer signature contains of two elements (σ', r), where one element is in $\mathbb{G}$ and the other element is in $\mathbb{Z}_q^*$. We can select the parameter such that the elements in $\mathbb{G}$ are 171-bits strings. Therefore, we obtain a designated confirmer signature whose length is 171+160=331 bits. However, to achieve such short size, currently, we can only use supersingular elliptic curve to implement ZCSM signature scheme.

6. CONCLUSION AND FURTHER WORKS

A new designated confirmer signature scheme is proposed in this paper. The new scheme is much more efficient than the previous schemes and does not need any commitment scheme or strong witness hiding proofs. For the further work, we will study how to construct the designated confirmer signature scheme which is secure against adaptive adversaries (Such model was considered in [6]) and the secure confirmer signature scheme under the new definitions proposed by [19] and [18] from the proposed DCS scheme in this paper.

7. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 60773202, No. 60503006 and No. 60633030) and 973 Program (2006CB303104).

8. REFERENCES

[1] M. Bellare and A. Palacio. GQ and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attack. In *Proceedings of Crypto 2002, volume 2442 of LNCS*, pages 162–177. Springer-Verlag, Aug. 2002.

[2] D. Boneh and X. Boyen. Short signatures without random oracles. In *Proceedings of Eurocrypt 2004, volume 3024 of LNCS*, pages 56–73. Springer-Verlag, May 2004.

[3] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of Crypto 2001, volume 2139 of LNCS*, pages 213–229. Springer-Verlag, Aug. 2001.

[4] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Proceedings of Asiacrypt 2001, volume 2248 of LNCS*, pages 514–532. Springer-Verlag, Dec. 2001.

[5] J. Camenisch and M. Michels. Efficient group signature schemes for large group. In *Proceedings of Crypto 1997, volume 1296 of LNCS*, pages 410–424. Springer-Verlag, Aug. 1997.

[6] J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. In *Proceedings of Eurocrypt 2000, volume 1807 of LNCS*, pages 243–258. Springer-Verlag, May 2000.

[7] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Proceedings of Crypto 2003, volume 2729 of LNCS*, pages 126–144. Springer-Verlag, Aug. 2003.

[8] D. Chaum and H. V. Antwerpen. Undeniable signatures. In *Proceedings of Crypto 1989, volume 435 of LNCS*, pages 212–216. Springer-Verlag, Aug. 1989.

[9] D. Chaum and H. V. Antwerpen. Designated confirmer signatures. In *Proceedings of Eurocrypt 1994, volume 950 of LNCS*, pages 86–91. Springer-Verlag, May 1994.

[10] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Proceedings of Crypto 1992, volume 740 of LNCS*, pages 89–105. Springer-Verlag, Aug. 1992.

[11] C. Gentry, D. Molnar, and Z. Ramzan. Efficient designated confirmer signatures without random oracles or general zero-knowledge proofs. In *Proceedings of Asiacrypt 2005, volume 3788 of LNCS*, pages 662–681. Springer-Verlag, Dec. 2005.

[12] S. Goldwasser and E. Waisbard. Transformation of digital signature schemes into designated confirmer signature schemes. In *Proceedings of TCC 2004, volume 2951 of LNCS*, pages 77–100. Springer-Verlag, March 2004.

[13] M. Michels and M. Stadler. Generic constructions for secure and efficient confirmer signature schemes. In *Proceedings of Eurocrypt 1998, volume 473 of LNCS*, pages 458–464. Springer-Verlag, May 1998.

[14] J. Monnerat and S. Vaudenay. Chaum's designated comfirmer signature revisited. In *Proceedings of ISC 2005, volume 3650 of LNCS*, pages 164–178. Springer-Verlag, Sep. 2005.

[15] T. Okamoto. Designated confirmer signatures and public-key encryption are equivalent. In *Proceedings of Crypto 1994, volume 839 of LNCS*, pages 61–74. Springer-Verlag, Aug. 1994.

[16] R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk. Universal designated-verifier signatures. In *Proceedings of Asiacrypt 2003, volume 2894 of LNCS*, pages 523–542. Springer-Verlag, Dec. 2003.

[17] G. Wang, J. Baek, D. S. Wong, and F. Bao. On the generic and efficient constructions of secure designated confirmer signatures. In *Proceedings of PKC 2007, volume 4450 of LNCS*, pages 43–60. Springer-Verlag, April 2007.

[18] V. K. Wei. Invisible designated confirmer signatures without random oracles. In *Proceedings of the 2nd*

ACM symposium on Information, computer and communications security - ASIACCS '07, pages 356–358. ACM Press, 2007.

[19] D. Wikström. Designated confirmer signatures revisited. In *Proceedings of TCC 2007, volume 4392 of LNCS*, pages 342–361. Springer-Verlag, March 2007.

[20] F. Zhang, X. Chen, W. Susilo, and Y. Mu. A new signature scheme without random oracles from bilinear pairings. In *Proceedings of VietCrypt 2006, volume 4341 of LNCS*, pages 67–80. Springer-Verlag, Sep. 2006.

[21] F. Zhang, R. Safavi-Naini, and W. Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In *Proceedings of Indocrypt 2003, volume 2904 of LNCS*, pages 191–204. Springer-Verlag, Dec. 2003.

[22] F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Proceedings of PKC 2004, volume 2947 of LNCS*, pages 277–290. Springer-Verlag, March 2004.