International Journal of Security and Its Applications
Vol.9, No.2 (2015), pp.169-178
http://dx.doi.org/10.14257/ijsia.2015.9.2.16

Research and Improvement of Ordered Multisignature Algorithm in Random Oracle Model

Yonglong Tang

School of mathematics and Statistics Jishou University
tyltyls@163.com

Abstract

Ordered multisignature allows signers attest to a common message as well as the order in whey they signed. Unlike multi-signature, aggregate signature aggregate signature scheme provides a method to aggregate signature by signature on different messages. In this paper, we presented an ordered multisignature provably secure without random oracles. We mainly focus on construction of perfectly hiding and computationally binding commitment (PHCBC). Our construction was based on the signature scheme of waters and is computationally suitable for practical application. Experiment showed that this scheme was suitable for the practical application with high computing efficiency. By transmitting individual signatures into a multi-signature, multi-signature scheme can greatly save communication cost.

Keywords: *multisignature, random oracles, signature scheme, Bilinear maps*

1. Introduction

Multisignature scheme allows different signers signing the same document, a multi-signature of roughly the same size as a standard signature. A verifier is convinced that each signer participated in signing by transmitting a multi-signature instead of individual signatures, multi-signature schemes can greatly save on communication costs. Multisignatures were first introduced by Itakura and Nakamura [1] and have been the topic of muchresearch [2-4]. Micali, Ohta and Reyzin [5] gave the first strong notion of security of multisignatures. Boldyreva [4] gave a variant of Micali-Ohta-Reyzin model. It is more general and solved the open problemstated [5]. Moreover, the scheme of Boldyreva is the first non-interactive.

Mihir Bellare and Gregory Neven [6] study an identity based multi-signature scheme from RSA. Craig and Zulfikar Ramzan [7] also study an identity-based multi-signature scheme. Multisignatures is related to the aggregate signature. Boneh,Gentry,Lynn and Shacham [8] defined an aggregate signature scheme. Unlike multi-signature, aggregate signature aggregate signature scheme provides a method to aggregate signature by signature on different messages. Alexandra Boldyreva *et al.* [9] propose a new primitive that they call ordered multi-signatures (OMS) and a formal security model for ordered multi-signatures. The ordered multisignaturesproduces a compact (sontant-size). Multi-signatures, uses constant-size keys, is sequential in that signers sign one after another and no further interaction among the signers is rquired and ensures authenticity of both the signing order and that of message [9]. They gave some potential applications of ordered multi-signatures scheme in more detail.

Comparing to the struct signers [10] ,the ordered multi-signatures scheme in [9] is in the noninteractive setting. Provable security is the basic requirement for signature schemes. Alexandra Boldyreva *et al.* [9] proved ordered multisignature scheme secure in the random oracle model [11]. Commitment scheme is a basic building block and has diverse applications

ISSN: 1738-9976 IJSIA

to cryptographic proto-cols, especially to zero-knowledge proofs [17]. Informally, a commitment scheme is a two stage protocol between a sender and a receiver. In the first stage, the sender commits to a value b, and in the second, the sender 'reveal' this value to the receiver. We want two security properties from a commitment scheme. The hiding property says that the receiver does not learn anything about the value b during the commit stage. And the binding property says that after the commit stage there is at most one value that the sender can successfully open. According to the computational power of senders and receivers, commitments can be classified into several possible types [18]. In this paper, we mainly focus on construction of perfectly hiding and computationally binding commitment (PHCBC).

Works about perfectly hiding commitment: Construction of PHCBC is an attractive problem. PHCBC s with a constant number of rounds were shown exist based on specific number-theoretic assumptions (more generally, based on any collection of claw-free permutations with an efficiently-recognizable index set), and collision-resistant hash function [19]. Protocols with higher round complexity were shown to exist based on different types one-way functions. Protocols with O (n/ log n) rounds were based on one-way permutations [20] and regular one-way functions [21]. Finally, a protocol with a polynomial number of rounds was based on any one-way function [22]. in (n/log n) rounds were shown to be the tight lower bound on the rounds complexity of PHCBC [23].

While, it is not known whether PHCBC in constant rounds constructed can be achieved with one-way function or one-way permutation.There are many so-called atomic protocols for NP that achieve constant error-probability in constant (three or four) moves [24]. Serial repetition lowers the error and preserves ZK, but at the cost of increasing the number of rounds to non-constant. So we would like to do parallel repetition. However, this is ruled out: first, we have the above mentioned results of [25]; secondly, the latter also showed that in general parallel repetition does not preserve ZK. So one must build low error ZK protocols directly [26].

Informally, an multi-signature.scheme allows different signers to jointly sign the same document, and yield a multi-signature with the same size as a standard signature. A verifier is convinced that each signer participates in signing. By transmitting individual signatures into a multi-signature, multi-signature scheme can greatly save communication cost. Boldyreva, provable security is the basic requirement for signature schemes. Alexandra Boldyreva *et al.* [27] proved that ordered multisignature scheme is secure in the random model. However, we find that their proof is wrong. This paper analyzes the reasons for the insecurity of the previous multi-signature scheme, and puts forward a genuine random sequential multi-signature scheme based on the waters signature scheme. Comparing to the structured signatures, the ordered multi-signatures scheme in is in the noninteractive setting. It is practical. Provable security is the basic requirement for signature schemes. Boldyreva, 2003. proved ordered multisignature scheme secure in the random. We give an ordered multisignature scheme that is provably secure without random oracles, by using a technique similar to tha of [12], thking inspiration from the method [9]. The paper analyzes the reasons for the insecurity of the previous multi-signature scheme, and puts forward a genuine random sequential multi-signature scheme based on the waters signature scheme, and the experiment proves that this scheme is a good scheme suitable for the practical application with high computing efficiency.

2. Digital Signature Schemes and Security Model

2.1. Digital Signature Schemes

A signature scheme consists of the following three oracle model. Algorithms is a key generation, signature generation algorithm sign and a signature verification algorithm Verify [14]. Key Gen, which on input 1^k , where k is the security parameter, outputs a pair (p_k , s_k) of matching public and private keys. Algorithm Key Gen is probabilistic, which receives a message m and the private key s_k , and outputs a signature

$$\theta = \text{Sign (m,} s_k) \qquad (1)$$

The signing algorithm might be probabilistic. Verify, which receives a candidate signature θ , a message m and a public key p_k , and returns an answer Verify (p_k , m, θ)

as to whether or not θ is a valid signature of m with respect to p_k . In general, the verification algorithm need not be probabilistic.Existential unforgeability under an adaptive chosen message attack introduced by Goldwasser, Micali and returns an answer Verify(p_k , m, θ) as to whether or not θ is a valid signature of m with respect to p_k . In general, the verification algorithm need not be probabilistic.valid signature of m with respect to p_k . In general, the verification algorithm need not be probabilistic.

Existential unforgeability under an adaptive chosen message attack introduced by Goldwasser, Micali and Rivest [15], has become the standard notion of signature security. The security model of existential unforgeability (in the random oracle model) is defined using the following game between a challenger and an adversary [16].

2.2. Security Model

Setup: The challenger runs algorithm key obtain to a public p_k and private key s_k . The adversary A is given p_k .Queries :Proceeding adaptively, A requests signatures with p_k on at most qs messages of his choice.

$$N_1 ,\ N_2 ,\ \ldots,\ N_{qs} \in \{0, 1\}^* \qquad (2)$$

The challenger responds to each query with a signature

$$\theta_i = \text{Sign (} s_k \text{, Mi)} \qquad (3)$$

Algorithm A also adaptively asks for at most q_H queries of the random oracle H. Eventually, A outputs a pair (M, θ) and wins the game if

1) M is not any of N_1 , N_2 , …, N_{qs} .
2) Verify (p_k , M, θ) = valid.

Protocol 1 Secure Euclidean Distance Computation Protocol

Input of server：Biometric feature vectors v={ v1,…vn}, key pair< pk_i , sk_i >

Input of client：Biometric feature vectors v'={v1',…vn'}, ciphertext under pk_i : [v']={[v1'],…[vn']}

Output of server：Euclidean distance between feature vectors d= $\sum_{j=1}^{n}(v_j - v_j')^2$

Output of client：none

The protocol execution process is as follows:

The server encrypts v={ v1,…vn} with pk as [v]={[v1],…[vn]}, and sends [v] to the client;

The client chooses random numbers r1,…rn, and encrypts them with pk as [r]={[r1],…[rn]}. Then the client computes [xj]=[vj-vj'+rj]=[vj]× [rj]/[vj']. The client sends [s] to the server where [s]={[s1],…[sn]}.

The server opens [x] which means decrypting the ciphertext with the private key and get s={s1,…sn}. Then the server computes yj=xj2 where j=1,…n, and encrypts y={ t1,…tn} with pk as [t]={[t1],…[tn]}. Then [y] is sent to the client.

The client computes rj2 and encrypts it as [rj2] where j=1,…n. Then the client computes [dj]=[tj-rj2-2rj(vj-vj')]= [tj]/([rj2] × (vj-vj')]2rj) where j=1,…n, and$[d]=[\sum_{j=1}^{n} d_j]=\prod_{j=1}^{n}[d_j]$.

Then [d] is sent to the server.

The server opens [d] and gets the Euclidean distance d between feature vectors. Output d.

3. Waters Signature Scheme and Ordered Multisignature Schemes

3.1. Bilinear Maps

G and GT are multiplicative cyclic groups of order p ; the group action on G and G_T can be computed efficiently; g is a generator of G :

$$\text{e: } G\times G \rightarrow G_T \tag{4}$$

Which is an efficiently computable map with the following properties:

Bilinear: for all u ,v ∈ G and a , b ∈ Z, e(u^a , v^b) = e $(u,v)^{ab}$;

2) Non-degenerate: e (g , g)≠ I_{Gt} , where I_{Gt} is the identity of G_T .

We say that G is a bilinear group if it satisfies these requirements.

3.2. The Waters Signature Scheme

The messages will be assumed to be bit strings of the form $\{0,1\}^k$. In practice, a collision-resistant hash function H_k :

{0, 1}*$\rightarrow \{0,1\}^k$ can be used to create message of the desired length.

The scheme first choose groups G and G_T of prime order p such that an admissible pairing:

e: $G\times G \rightarrow G_T$ can be constructed and choose a random Generator. g∈G , k+1 additional random generators :

$$u', u_1, u_2, \dots, u_k \in \quad \text{G} \tag{5}$$

The Waters signature scheme consists of three algorithms as follows Key Gen. Pick random

$$\alpha \ \leftarrow_R Z_p \tag{6}$$

and set A← e $(g,g)^{\alpha}$.

The public key p_k is A ∈ G_T .

The private key s_k is α .Sign(s_k , m). Parse the user's private key s_k as α ∈ Z_p and the message m as a bitstring

$$(n_1, n_2, \dots, n_k) \in \{0,1\}^k \tag{7}$$

Pick a random r $\in {}_R Z_p$ and compute

$$\theta_1 \leftarrow g^{\alpha}\left(u'\prod_{i=1}^{k}u_i^{n_i}\right)^T \tag{8}$$

$$\theta_2 \leftarrow g^T \tag{9}$$

The signature is $\theta = (\theta_1, \theta_2) \in G^2$

Verify (p_k , m, θ). Parse the user's public p_k as A $\in G_T$, the message m as a bitstring:

$$(n_1, n_2, \dots, n_k) \in \{0,1\}^k \tag{10}$$

and the signature θ as $(\theta_1, \theta_2) \in G^2$. Verify that:

$$e(\theta_1, g) \bullet e\left(\theta_2, u'\prod_{i=1}^{k}u_i^{n_i}\right)^T = \mathrm{A} \tag{11}$$

This signature is existentially unforgeable under an adaptive chosen message attack if the Computational Diffie-Hellman (CDH) problem[13] in bilinear groups is hard.

3.3. Ordered Multisignature Schemes and their Security

An ordered multi-signature scheme (OMS) consists of the following four algorithms [31]. A parameter generation algorithm Opg that returns some global information I for the scheme. This algorithm can be run by a trusted third-party or standards bodies. A key generation algorithm Okg run by a user that on the input global information I returns a publicprivate key-pair (p_k , s_k). A signing algorithm OSign run by a user on inputs its secret key s_k , a message n$\in$ {0, 1}*, a list of i-1 public keys

$$\mathrm{L} = (\mathrm{p}k_1, \dots, \mathrm{p}k_{i-1}) \tag{12}$$

An OMS-so-far θ'. It returns a new OMS θ, or $\perp$ if the input is deemed invalid. A deterministic verification algorithm OVf that on inputs a list of public keys (pk_1,,…, pk_n), a publicprivate key-pair (p_k , s_k). A signing algorithm OSign run by a user on inputs its secret key s_k , a message n$\in$ {0, 1}*, a list of i-1 public keys:

$$\mathrm{L} = (\mathrm{p}k_1, \dots, \mathrm{p}k_{i-1}) \tag{13}$$

The OMS-so-far θ'. It returns a new OMS θ, or $\perp$ if the input is deemed invalid. A deterministic verification algorithm OVf that on inputs a list of public keys (pk_1,,…, pk_n), a message n, and an OMS θ returns valid or $\perp$.

The security model of OMS in [6] extends the notion of security for multi-signatures in [8] to also ensure authenticity of the signing order. Similarly to the model of [8], the users are required to prove knowledge of their secret keys during public-key registration with a CA. For simplicity, this is modeled by requiring an adversary to hand over secret keys of malicious signers. This is known as theregistered-key or certified-key model. The security model of existential unforgeability for OMS is defined using the following game associated to OMS and a forger A with access to an oracle. The game runs in three stages[11]:

The game first runs Opg to obtain output I and then generates a challenge key-pair (pk, sk) by running Okg on input I[13]. Attack: A runs on inputs I, pk. A may query a key registration oracle with a key-pair (pk', sk') and coins c used for key generation , which records pk' as registered if Okg(I,c) $\Rightarrow (pk', sk')$ (This is a simplified model of a possibly morecomplex key registration protocol with a CA that involves proofs of knowledge of secret keys.) A also has access to a signing oracle OSign, which on inputs m, θ , L returns $\perp$ if not all public keys in L are registered and OSign(s_k , m, θ, L) otherwise.

Forgery: Eventually, A halts with outputs a list of public keys

$$L^* = (pk_1 *, \dots , pk_n *) \tag{14}$$

The message n*, and a purported OMS signature . This output is considered to be a forgery if it holds that[11].

1) OVf (L*, n*, θ *) = valid;

2) pk_{i^*} *= pk for some i*$\in$ {1, … , n};

3) All public keys in L*except pk are registered;

4) A did not query, n*, θ', L' to its signing oracle where $|L'|$ = i*-1, for any $\theta' \in$ {0, 1}*.

We define that an ordered multi-signature is (t, q_c , q_s ,N , ε) unforgeable if not t-time adversary making q_c certification queries and q_s signing queries can win the above game with advantage more than ε, where N is an upper bound on the length of the sequential signatures involved

4. The Proposed Scheme

We construct an ordered multi-signature WOMS from the Waters signature. Our scheme is defined by the following algorithms shown in Figure 1. The messages will be assumed to be bit strings of the form $\{0,1\}^k$ .Parameter generation algorithm and key generation algorithm were produced as the Waters signature scheme. Parameter generation Opg: The algorithm first choose groups G and GT of prime order p such that an admissible pairing e:

$$G \times G \rightarrow G_T \tag{15}$$

can be constructed and choose a random generator g$\in$G, k+1 additional random generators:

$$u' , u_1 , u_2 , \dots , u_k \in \mathrm{G} \tag{16}$$

Key genetation Okg: Pick random $\alpha \leftarrow_R Z_p$ and set A$\leftarrow$ e$(g,g)^\alpha$. The public key pk is A$\in G_T$. The private key s_k is θ .Signing OSign: On inputs ski:

$$\mathrm{n}=(n_1 ,\ n_2 , \dots ,\ n_k) \in \{0,1\}^k \tag{17}$$

$$L = (pk_1 ,\dots , pk_{i-1}) \tag{18}$$

An OMS-so-far θ' ,the algorithm first verifies that OVf (L, m, θ')= valid, as defined below and if not, outputs $\perp$ and halt. For a first signer, θ' is defined as (I_G , I_G , I_G). Then parse θ' as (S' , R' , T')$\in$G3. Choose random $r_i , t_i \in Z_p$, and compute:

$$\mathrm{S} = S'g^{\alpha_i + it_i}\left(u'\prod_{j=1}^{k} u_j{}^{n_j}\right)^{T_j} \tag{19}$$

$$R = R'g^{T_i} ,\ T = T'g^{it_j} \tag{20}$$

The signature is θ = (S, R, T). Verification Ovf: On inputs (pk_1 ,… , pk_n), n=(n_1, n_2, … , n_k)$\in \{0,1\}^k$, θ , the algorithm first checks that all pk_1 ,… , pk_n are distinct, if not, it output $\perp$ and halt. Then parse θ as (S, R, T) and verify if

$$e\left(R, u'\prod_{j=1}^{k} u_j^{n_j}\right)^{-1} = A_1\ A_2 \dots A_n \text{ e (T, g)} \tag{21}$$

If so, output valid ; if not, output $\perp$. An ordered multisignature in our scheme has the form

$$\text{S=} g^{\sum_{i=1}^{n}\alpha_i + it_i}\left(u'\prod_{j=1}^{k} u_j^{n_j}\right)^{\sum_{j=1}^{n} r_j} \tag{22}$$

$$\text{R=} g^{\sum_{i=1}^{n} r_i},\ \text{T=} g^{\sum_{i=1}^{n} it_i} \tag{23}$$

Correctness: It is easy to see that the verification equation is satisfied:

$$e\left(R, u'\prod_{j=1}^{k} u_j^{n_j}\right)^{-1} = A_1\ A_2 \dots\dots A_n \text{ e (T, g) } e\left(R, u'\prod_{j=1}^{k} u_j^{n_j}\right)\ e\left(R, u'\prod_{j=1}^{k} u_j^{n_j}\right)^{-1} \tag{24}$$

$$\text{Let } T_i = e(g,g)^{t_i}\text{, then e(T, g) = } e(g^{\sum_{i=1}^{n} it_i}, g) = e(g\ ,g)^{\sum_{i=1}^{n} it_i} = \prod_{i=1}^{n} T_i^i \tag{25}$$

The equation above ensures authenticity of the signing order.

5. The Security Proof

The security analysis of our scheme is similar to the analysis presented in[13]. Theorem: The WOMS is (t, q_c, q_s ,N , ε)-unforgeable if the Waters signature scheme is (t, q_c, q_s ,N , ε) -unforgeable on G, where

$$t' = \text{t +O(} q_c + N_{q_s} \text{ +N)} \tag{26}$$

$$q' = q_s\ ,\ \varepsilon' = \varepsilon \tag{27}$$

Proof: Suppose A is a forger algorithm that (t, q_c, q_s ,N , ε)-breaks our WOMS. We construct an algorithm B that (t', q' , ε')-breaks the Waters signature scheme. Algorithm B is given a public key of the Waters signature scheme, A = e$(g,g)^{\alpha}$. It interacts with A as follows.

Setup: Algorithm B runs A supplying it with the challenge key

$$p_k = \text{A = e}(g,g)^{\alpha} \tag{28}$$

Certification Queries: A wish to certify some public key pk', providing also its corresponding private key s. Ak'lgorithm B checks that the private key is correct and if so, registers (pk', sk') in its list of certified key pairs. OMS Signature Queries: Algorithm A requests an OMS under the challenge key p_k on a message m. In addition, it supplies an OMS-so-far θ', a list of i-1 public keys:

$$\text{L = (p}k_1,\dots,\text{p}k_{i-1}) \tag{29}$$

The simulator B first checks that the signature θ' is valid; that each key in L has been certified; that the challenge key does not appear in L; and that |L| < N. B returns $\perp$ if any of these conditions does not hold. Otherwise, B queries its own signing oracle for key p_k , obtaining a signature θ on message n.

$$\theta_1 = g^{\alpha_j} \left(u' \prod_{j=1}^{k} u_j^{m_j} \right)^{T_j} \tag{30}$$

$$\theta_2 = g^{T_j} \tag{31}$$

B parse θ' as ($S_{i-1}, R_{i-1}, T_{i-1}$), B pick a random $t_i\ {}_R Z_p$, and compute $S_i = S_{i-1}\ \theta_1\ g^{it_i}$.

$$g^{\sum_{h=1}^{i} \alpha_h + ht_h} \left(u' \prod_{j=1}^{k} u_j^{m_j} \right)^{\sum_{h=1}^{i} r_h} = \theta = \left(S_i, R_i, T_i \right) \tag{32}$$

The OMS on message m under keys

$$\text{L} = (\text{p}^{k_1}, \ldots, \text{p}^{k_{i-1}},\ p_k) \tag{33}$$

Output: Eventually, A halts, outputting a forgery θ *= (S*, R*, T*), a message.

$$\text{n}^* = (n_1{}^*,\ n_2{}^*, \ldots,\ n_k{}^*) \in \{0,1\}^k \tag{34}$$

and a list of public keys

$$\text{L}^* = (\text{p}^{k_1 *}, \ldots, \text{p}^{k_n *}) \tag{35}$$

This forgery must verify as valid under Ovf; all public key in L* except p_k must have been certified;

$$\text{p}^{k^*_{i^*}} = p_k \tag{36}$$

for some i*∈{1, … , n}; |L*| ≤ N; and A did not query m*, θ', L' to its signing oracle where |L′| = i*-1. For any θ' ∈ {0, 1}*, where S*, R*, T* is as follows.

$$\text{S}^* = g^{\sum_{i=1}^{n} \alpha_i} \left(u' \prod_{j=1}^{k} u_j^{m^*{}_j} \right)^{\sum_{i=1}^{n} r_i} \tag{37}$$

$$\text{R}^* = g^{\sum_{i=1}^{n} r_i}, \quad \text{T}^* = g^{\sum_{i=1}^{n} t_i} \tag{38}$$

Now, Algorithm B computes

$$\theta_1 = \text{S}^* g^{\alpha_j} \prod_{i \neq i^*} \left(g^{\alpha_i} \right)^{-1} \left(T^* \right)^{-1} \tag{39}$$

$$\theta_2 = \text{R}^*,\ \alpha_i (\ i \neq i') \tag{40}$$

The private key corresponding to each public key in L* , B can knows it by the certification procedure. We have e(θ_1 ,g)

$$\text{e}\left(\theta_2, u' \prod_{j=1}^{k} u_j^{m^*{}_j} \right)^{-1} = \text{e(S*,g) .e}\left(\prod_{i \neq i^*} g^{\alpha_i}, g \right)^{-1} \tag{41}$$

$$\text{e}\left(R^*, u' \prod_{j=1}^{k} u_j^{m^*{}_j} \right)^{-1} = \text{e(S*,g)} \tag{42}$$

$$\mathrm{e}\left(T^*,g\right)^{-1}\prod_{i\neq i^*}e\left(g,g\right)^{-\alpha_i}=A_1\,A_2\ldots\ldots A_n\ (41)\ \mathrm{e}\ (T^*,\mathrm{g})^{-1}.\ \prod_{i\neq i^*}A_i^{-1}=A_{i^*} \qquad (43)$$

So θ_1, θ_2 is a valid Waters signature on n* under the challenge key $p_k = A_{i^*}$, Since A did not make an OMS signing query at n*, B did not make a signing query at n*, so $\theta =$ θ_1 , θ_2 is a nontrivial Waters signature forgery. Algorithm B outputs $\theta =$ θ_1, θ_2 and halts.

Algorithm B succeeds whenever A does. The running-time of B includes:

1) B's signing queries. B makes as many signing queries as A makes OMS signing queries;

2) B handles A's certification queries. Each certification query can be handled in O(1) time;

3) B handles OMS signing queries. Each OMS signing query can be handled in O(N) time;

4) The other computations can be completed in O(N) time.

6. Conclusions

A PHCBC in two rounds from any one-way permutation is a negation of the result. Σ protocol is our main tool to construct PHCBC. Σ protocol is a three-move interactive protocol between the prove and the verifier in which the verifier is only required to send random bits as a challenge to the prove. Based on Σ protocol, a new method to construct a commitment scheme was proposed. In this paper, we will use Σ-protocol on Hamiltonian-Cycle to construct PHCBC in constant rounds from any one-way permutation. In this paper we gave an ordered multisignature scheme which is provably secure without random oracles. Our construction derives from the waters signature scheme. It is also an interesting problem to find an ordered aggregate signature scheme provable secure without random oracles. and the experiment proves that this scheme is a good scheme suitable for the practical application with high computing efficiency.

References

[1] K. Itankura and K. Nakamura, “A public-key crptosysterm suitable for digital multisignatures”, NEC J. Res. & Dev., vol. 71, **(1983).**

[2] M. Bellare and G. Neven, “Identity-Based Multisignatures from RSA”, In CT-RSA, 2007, LNCS p. 4377**, (2007).**

[3] C. Gentry and Z. Ramzan, “Identity-Based Aggregate Signatures”, In PKC 2006, LNCS, 3958, **(2006).**

[4] D. Boneh, C. Gentry, B. Lynn and H. Shacham, “Aggregate and verifiably encrypted signatures from bilinear maps”, In Proceedings of Euro-crypt 2003, LNCS, 2656, **(2003).**

[5] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham and B. Waters, “Sequential Aggregate Signatures and Multisignatures without Random Oracles”, In EUROCRYPT 2006, LNCS, 4004, **(2006).**

[6] C. M. Tang, D. Y. Pei, Z. J. Liu and X. F. Wang, “Delegateable Signature Based on Non-interactive Witness Indistinguishable and Non-interactive Witness Hiding Proofs”, Science in China Series F: Information Sciences, **(2008),** pp. 68-78.

[7] C. Y. Lin, T. C. Wu and F. Zhang, “A Structured, “Multisignature Scheme from the Gap Difffie-Hellman Group”, Cryptology ePrint Archive, Report, vol. 90, **(2003).**

[8] I. Haitner, J. J. Hoch, O. Reingold and G. Segev, “Finding Collisions in Interactive Proto-cols - A Tight Lower Bound on the Round Complexity of Statistically-Hiding Commitments”, **(2008).**

[9] K. Ohta and T. Okamoto, “Multisignature schemes secure against active insiderattacks”, IEICE Trans. Fundamentals, E82-A, vol. 1, **(1999).**

[10] T. Okamoto, “A digital multisignature scheme using bijective public-key cryptosystems”, ACM Trans. Computer Systems, ACM Press, New York, vol. 6, no. 4, **(1988).**

[11] A. Boldyreva, “Threshold signature, multisignature and blind signature schemes based on the gap-Diffie-Hellmangroup signature scheme”, In Proceedings of PKC 2003, LNCS, 2567, **(2003).**

[12] S. Micali, K. Ohta, and L. Reyzin, “Accountable-subgroup multisignatures (extended abstract)”, In Proceedings of CCS 2001, ACM Press, **(2001).**

[13] S. Goldwasser, S. Micali, and R. Rivest, “A digital signature scheme secure against adaptive chosen-message attacks”, SIAM J. Computing, **(1988).**

[14] H. V. Jansen, N. R. Tas and J. W. Berenschot, “in Encyclopedia of Nanoscience and Nanotechnology, Edited H. S. Nalwa, American Scientific Publishers, Los Angeles, vol. 5, **(2004)**, pp. 163-275.

[15] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung, “Perfect Zero-Knowledge Arguments for NP Using AnyOne-Way Permutation”, J. Cryptology, vol. 11, no. 2, **(2000),** pp. 87-108.

[16] M. Nguyen and S. Vadhan, “Zero-Knowledge with Efcient Provers”, In Proc. 38th STOC, vol. 6, **(2006),** pp 287-295.

[17] D. Catalano and I. Visconti, “Hybrid Commitments and Their Applications to Zero-knowledge Proof Systems”, Theoretical Computer Science, vol. 10, **(2007),** pp. 229-260.

[18] O. Goldreich, “Foundations of Cryptography (Basic Tools)”, Cambridge University Press, vol. 8, **(2001),** pp. 68-88.

[19] J. Kimura and H. Shibasaki, “Recent Advances in Clinical Neurophysiology”, Proceedings of the 10th International Congress of EMG and Clinical Neurophysiology, **(1995)** May 15-19; Kyoto, Japan

[20] A. Boldyreva, C. Gentry, A. O’Neill and D. H. Yum, “Ordered Multisignatures and Identity-BasedSequential Aggregate Signatures with Applications to Secure Routing”, In Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM Press, **(2001)** October, pp. 216–225; New York.

[21] C. Y. Lin, T. C. Wu and F. Zhang, “A Structured Multisignature Scheme from the Gap Difffie-Hellman Group”, Cryptology ePrint Archive, Report **(2003)** October 15-19; Kyoto, Japan

[22] M. Bellare and P. Rogaway, “Random oracles are practical: A paradigm for designing efficient protocols”, In ACM CCS, vol. 93, ACM Press, (**1993**) May 62-73, New York.

[23] B. Waters, “Effcient identity-based encryption without random oracles”, In Proceedings of Eurocrypt 2005, LNCS 3494, **(2005)** October 14-27 Springer Berlin.

[24] J. Stern, D. Pointcheval, J. Malone-Lee and N. P. Smart, “Flaws in Applying Proof Methodologies to Signature Schemes”, In CRYPTO 2002, LNCS 2442, **(2005)** October 14-27, Springer Berlin.

[25] S. Goldwasser, S. Micali, and R. Rivest, “A digital signature scheme secure against adaptive chosen-message attacks”, SIAM J. Computing, vol. 17, no. 2, **(2000)** May , pp. 281-308, Springer Berlin.

[26] A. Lysyanskaya, S. Micali, L. Reyzin and H. Shacham, “Sequential Aggregate Signatures from Trapdoor Permutations”, In EUROCRYPT 2004, LNCS, 3027, **(2004)** October 74-90, Springer Berlin.

[27] A. Lysyanskaya, S. Micali, L. Reyzin and H. Shacham, “Sequential Aggregate Signatures from Trapdoor Permutations”, In EUROCRYPT 2004, LNCS 3027, **(2004)** October 74-90, Springer Berlin.

[28] A. Boldyreva, C. Gentry, A. O’Neill and D. H. Yum, “Ordered Multisignatures and Identity- Based Sequential Aggregate Signatures with Applications to Secure Routing”, In Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM Press, **(2007)** May, pp. 276–285, New York.

[29] C. M. Tang, D. Y. Pei and Z. A. Yao, “E±cient Zaps and Signatures of Knowledges”, In Proceeding of IEEE International Conference on Computational Intelligence and Security (CIS'2007), **(2008)** October, pp. 637-641, Kyoto, Japan.

Author

Yonglong Tang, was born on May 20, 1961, in Zhangjiajie of Hunan province, and graduated from Huan Normal University on July,1982 with a mathematics bachelor degree. And he acted as a visiting fellow in Zhongshan University in 2007. Now he is an associate professor in college of mathematics and statistics in Jishou University with an annual income of 100,000 yuan. And his main research fields include applied mathematics and cryptology.