Buletin Teknik Elektro dan Informatika
(Bulletin of Electrical Engineering and Informatics)
Vol. 3, No. 1, March 2014, pp. 55~68
ISSN: 2089-3191

A Genuine Random Sequential Multi-signature Scheme

Yonglong Tang
School of Mathematics and Statistics, Jishou University
email: tyltyls@163.com

Abstract

The usual sequential multi-signature scheme allows the multi-signers to sign the document with their own information and sequence, and the signature is not real random and secure. 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.

Keywords: *bilinear maps, the waters signature scheme random, ordered multi-signature schemes*

1. Introduction

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

A multi-signature.scheme allows n different Signers to jointly sign the same document, yielding 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 n individual signatures, multi-signature schemes can greatly save on communication costs.

Mihir Bellare and Gregory Neven[1] for an Identity based multi-signature scheme from RSA. Craig and Zulfikar Ramzan[2] also for an identity-based multi-signature scheme. Alexandra Boldyreva et al.[5] propose a new primitive that they call ordered multi-signatures and a formal security model for ordered multi-signatures. “The ordered multi-signatures produces a compact multi-signatures, uses constant-size keys, is ‘sequential’ in that signers sign one after another and no further interaction among

Comparing to the “structured” signatures [13], the ordered multi-signatures scheme in [5] is in the noninteractive setting. It is practical. Provable security is the basic requirement for signature schemes. Alexandra Boldyreva et al.[5] proved ordered multisignature scheme secure in the random

Alexandra Boldyreva et al.[3] propose a new primitive that they call ordered multi-signatures and a formal security model for ordered multi-signatures. “The ordered multi-signatures produce compact multi-signatures use constant-size keys, are ‘sequential’ in that signers sign one after another and no further interaction among Comparing to the “structured” signatures [4], the ordered multi-signatures scheme in [3] is in the noninteractive setting. It is practical.

Provable security is the basic requirement for signature schemes. Alexandra Boldyreva et al.[3] proved ordered multisignature scheme secure in the random 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

Commitment scheme is a basic building block and has diverse applications to cryptographic proto-cols, especially to zero-knowledge proofs[14]. 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

Received November 23, 2013; Revised January 13, 2014; Accepted February 14, 2014

open. According to the computational power of senders and receivers, commitments can be classified into several possible types [15]. In this paper, we mainly focus on construction of perfectly hiding and computationally binding commitment (PHCBC).

Some 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 (or, more generally, based on any collection of claw-free permutations with an efficiently-recognizable index set), and collision-resistant hash function [16]. 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 [17] and regular one-way functions [18]. Finally, a protocol with a polynomial number of rounds was based on any one-way function [19]. in [20]O (n / log n) rounds were shown to be the tight lower bound on the rounds complexity of PHCBC.

The Problem: 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" ZK protocols for NP that achieve constant error-probability in constant (three or four) moves [22]. 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 [23]; second, the latter also showed that in general parallel repetition does not preserve ZK. So one must build low error ZK protocols directly [24].

2. Preliminaries

2.1. NP Relations

NP relations We say that a binary relation

$$R \subseteq \{0,1\}^* \rightarrow \{0,1\}^*$$

is an NP relation if there exists a polynomial p(.)such that for any

$$(x, w) \in R,$$

$$|w| \leq p(|x|)$$

and in addition there exists a polynomial time Turing machine for deciding membership in R. We denote by L_R the following:

$$L_R = \{x \mid \exists w s.t. (x, w) \in R\}$$

We say that

$$L \in NP \text{ if } L = L_R$$

for some NP relation R.

A negligible function is a function that grows slower that inverse of any polynomial. That is,

$$\nu : N \rightarrow N \circ$$

is negligible if for any positive polynomial p(¢) there exists a number n_0 such that

$$\nu(n) < 1/p(n)$$

for all

$$n > n_0$$

One-Way Function A function

f : $\{0,1\}^* \rightarrow \{0,1\}^*$

is called one-way if the following conditions hold:

1. There exists a deterministic polynomial-time algorithm A such that on input x, A outputs f(x);
2. For every non-uniform probabilistic polynomial-time algorithm A' there exists a negligible function ν such that for all sufficiently large k, it holds that

Prob (x ← $\{0,1\}^*$

$$A'(f(x)) \in f^{-1}(f(x)) < \nu(k)$$

f is a one-way permutation if it is a 1-1 and length preserving one-way function.

2.2. Commitment scheme

Definition 1 (Gen, Com, Ver) is a commitment scheme[11] if:

a. Efficiency: Gen, Com and Ver are polynomial-time algorithms;

b. Completeness: for all m it holds that

Prob (crs ← Gen (1^k);

(com, de c)←Com(crs , m);

Ver (crs, com, de c , m)=1)=1

c. Binding: for any polynomial-time algorithm sender there is a negligible function ° such that for all sufficiently large k it holds that

Prob(crs ← Gen (1^k)

(com, m_0 , m_1 ,de c_0 ,de c_1) ←Sender (crs)

$m_0 \neq m_1$ and

Ver(crs, com, d c_0 , m_0), Ver(crs,com,dec1, m_1)=1)≤ ν (k)

d. Hiding: for any adversary receiver there is a negligible function ° su that for all m_0 ; m_1 where

$m_0 = m_1$

and all sufficiently large k it holds that

Prob(crs←Gen(1^k);

b←{0,1};(com, de c)←Crs(crs, m_b):

b ← receiver (com))<1/2 + ν (k)

The "hiding" and "binding" of the above definitions are "computationally hiding" and "computationally binding" respectively A commitment is perfectly hiding if for any computationally unbounded adversary receiver for all m_0 ; m_1
where

$m_0 = m_1$

and all sufficiently large k it holds that

Prob(crs← Gen (1^k);

b←{0,1};(com, de c)←Crs(crs, m_b)

b← receiver (com))<1/2

A commitment is perfectly binding for any computationally unbounded sender for all sufficiently large k it holds that

Prob(crs← Gen (1^k);

(com, m_0 , m_1 ,de c_0 ,de c_1) ←Sender (crs):

$m_0 \neq m_1$

and

Ver (crs , com, de c_0 , m_0), Ver(crs,com,dec1, m_1)=1)=0

3. Digital Signature Schemes and Security Model

3.1. Digital Signature Schemes

A signature scheme consists of the following three oracle model[6]. algorithms: a key generation algorithm Key Gen, a signature generation algorithm Sign and a signature verification algorithm Verify[7].

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.
Sign, which receives a message m and the private key s_k , and outputs a signature

θ = Sign (m, s_k).

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 [9], 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 A [5].

3.2. Security Model

Setup the challenger runs algorithm Key Gen to obtain 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

$$N_1,\ N_2,\ \ldots,\ N_{qs} \in \{0, 1\}^* .$$

The challenger responds to each query with a signature θ_i □= Sign (s_k , Mi). Algorithm A also adaptively asks for at most q_H queries of the random oracle H.

Output : Eventually, A outputs a pair (M, θ □) and wins the game if

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

4. Preliminaries

4.1. Bilinear Maps [13].

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 ;

$$e: G \times G \to G_T$$

is an efficiently computable map with the following properties:

1) Bilinear: for all u ,v ∈ G and a , b ∈ Z,

$$e(u^a, v^b) = e\ (u,v)^{ab};$$

2) Non-degenerate:

$$e\ (g\ ,\ g) \neq I_{Gt}\ □□,$$

where I_{Gt} is the identity of G_T. We say that G is a bilinear group if it satisfies these requirements.

4.2. 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 :

$$\{0, 1\}^* \to \{0,1\}^k$$

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

$$e: G \times G \to G_T$$

can be constructed and choose a random Generator

$g \in G$, k+1

additional random generators

$u', u_1, u_2, \ldots, u_k \in$ □G.

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

$\alpha \leftarrow_R Z_p$

and set

A$\leftarrow$ e$(g, g)^{\alpha}$.

The public key p_k is

A $\in G_T$.

the private key s_k is α .Sign(s_k , m). Parse the user's private key s_k as

$\alpha \in Z_p$

and the message m as a bitstring

$(n_1, n_2, \ldots, n_k) \in \{0,1\}^k$,

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

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

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

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

$(n_1, n_2, \ldots, n_k) \in \{0,1\}^k$,

and the signature □ θ as

$(\theta_1,\theta_2)\in G^2$ □.

Verify that

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

holds if so, output valid ; if not, output invalid. This signature is existentially unforgeable under an adaptive chosen message attack if the Computational Diffie-Hellman (CDH) problem[13] in bilinear groups is hard.

4.3. Ordered multisignature schemes and their security

An ordered multi-signature scheme (OMS) consists of the following four algorithms[6]. 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

L = (p k_1 ,… , p k_{i-1}),

and 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 (p k_1 ,,…, p k_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

L = (p k_1 ,… , p k_{i-1}),

and 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 (p k_1 ,,…, p k_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 the registered-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]:

Setup the game first runs Opg to obtain output I and then generates a challenge key-pair (pk, sk) by running Okg on input I[3]. 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 more-complex 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*, … , pk_n*),

a 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*∈{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

θ' ∈{0, 1}*.[12]

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].

5. New Scheme

5.1. Construction

We construct an ordered multi-signature WOMS from the Waters signature. Our scheme is defined by the following algorithms. 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:

$G \times G \to G_T$

can be constructed and choose a random generator

g∈G, k+1

additional random generators

u', u_1, u_2, … , u_k ∈G.

Key genetation Okg: Pick random

$\alpha \leftarrow_R Z_p$

and set

A← e$(g,g)^\alpha$.

The public key pk is

A∈ G_T.

The private key s_k is θ .Signing OSign: On inputs ski,

n=(n_1 , n_2 , … , n_k)$\in \{0,1\}^k$,

L = (p k_1 ,… , p k_{i-1}),

and 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

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

$R = R' g^{T_i}$,

$T = T' g^{it_j}$

the signature is

θ □= (S, R, T).

Verification Ovf: On inputs

(p k_1 ,… , p k_n),

n=(n_1 , n_2 , … , n_k)$\in \{0,1\}^k$, θ □□,

the algorithm first checks that all p k_1 ,… , p k_n are distinct, if not, it output $\perp$ □and halt. Then parse θ □as (S, R, T) and verify if

e □S , g□□ $e\left(R, u' \prod_{j=1}^{k} u_j^{n_j} \right)^{-1}$

= A_1 A_2 … A_n e (T, g)

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

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

R= $g^{\sum_{i=1}^{n}r_i}$,

T= $g^{\sum_{i=1}^{n}it_i}$

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

e □S , g□□ $e\left(R,u'\prod_{j=1}^{k}u_j^{\,n_j}\right)^{-1}$

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

= $A_1\ A_2 \ldots\ldots A_n$ e (T, g)

Let $T_i = e(g,g)^{t_i}$,

Then

e(T, g) = $e(g^{\sum_{i=1}^{n}it_i},g)$

= $e(g\ ,g)^{\sum_{i=1}^{n}it_i}$

=$\prod_{i=1}^{n}T_i^{\,i}$

The equation above ensures authenticity of the signing order.

5.2. 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' = t +O($q_c + N_{q_s}$ +N),

q' = q_s ,

ε' = ε.

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 = A

= e$(g,g)^{\alpha}$

Certification Queries: A wish to certify some public key pk', providing also its corresponding private keys. 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

L = (pk_1,… , pk_{i-1}).

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.

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

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

B parse θ' □as ($S_{i-1}, R_{i-1}, T_{i-1}$), B pick a random $t_i\ {}_R Z_p$, and compute

$$S_i = S_{i-1}\ \theta_1\ g^{it_i}$$

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

θ = (Si, Ri, Ti)

is a OMS on message m under keys

L = (pk_1 ,… , pk_{i-1}, p_k)

Output Eventually, A halts, outputting a forgery

θ*= (S*, R*, T*),

a message

n*=(n_1*, n_2*,…, n_k *)∈ $\{0,1\}^k$, and a list of public keys

L* = (pk_1* ,… , pk_n*)

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

p$k^*_{i^*}$ = p_k

for some

i*$\in${1, … , n};

|L*| ≤□N;

and A did not query m*, θ', L' □to its signing oracle where

|L'| = i*-1

for any

θ' $\in${0, 1}*.

where S*, R*, T* is as follows.

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

R*= $g^{\sum_{i=1}^{n}r_i}$,

T*= $g^{\sum_{i=1}^{n}t_i}$

Now, Algorithm B computes

θ_1 = S* $g^{\alpha_j}\prod_{i\neq i^*}\left(g^{\alpha_i}\right)^{-1}\left(T^*\right)^{-1}$,

θ_2 = R*

α_i($i \neq i'$)

is the private key corresponding to each public key in L* , B can knows it by the certification procedure. We have

e(θ_1 ,g)

$$\text{e}\left(\theta_2, u'\prod_{j=1}^{k} u_j^{m^*{}_j}\right)^{-1} = \text{e(S*,g) .e}\left(\prod_{i\neq i^*} g^{\alpha_i}, g\right)^{-1} \text{.e}\left(T^*, g\right)^{-1}$$

$$\text{e}\left(R*, u'\prod_{j=1}^{k} u_j^{m^*{}_j}\right)^{-1} = \text{e(S*,g) . e}\left(R*, u'\prod_{j=1}^{k} u_j^{m^*{}_j}\right)^{-1} \text{.e}\left(T^*, g\right)^{-1} \prod_{i\neq i^*} e\left(g,g\right)^{-\alpha_i}$$

$$= A_1\, A_2 \ldots\ldots A_n \text{ e (T*,g). e } (T*, \text{g})^{-1}.\ \prod_{i\neq i^*} A_i^{-1}$$

$$= A_{i^*}$$

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

θ =□ θ_1 ,□

θ_2 is a nontrivial Waters signature forgery. Algorithm B outputs

θ =□ 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. Conclusion

We will construct a PHCBC in two rounds from any one-way permutation, which is a negation of the result in [9]. Σ protocol is our main tool to construct PHCBC. Σ protocol is a three-move interactive protocol between the prover and the verifier in which the verifier is only required to send random bits as a challenge to the prover . Based on Σ protocol, a new method to construct a commitment scheme was proposed in [10]. In this paper, we will use Σ-protocol on Hamiltonian-Cycle to construct PHCBC in constant rounds (i.e., two 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.

Acknowledgment

A Project Supported by Scientific Research Fund of Hunan Provincial Education Department (No. 09C791). Thank Wei-hua Hou.

References

[1] Mihir Bellare, Gregory Neven. *Identity-Based Multisignatures from RSA. In CT-RSA, LNCS 4377*, Springer, Berlin, 2007; 145–162.

[2] Jacques Stern, David Pointcheval, John Malone-Lee, Nigel P Smart. *Flaws in Applying Proof Methodologies to Signature Schemes . In CRYPTO, LNCS 2442*, Springer, Berlin. 2002; 93–110.

[3] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin. Hovav Shacham. Sequential Aggregate Signatures from Trapdoor Permutations. In EUROCRYPT, LNCS 3027, Springer, Berlin. 2004; 74–90.

[4] Alexandra Boldyreva. Craig Gentry, Adam O'Neill and Dae Hyun 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, New York, 2007; 276–285.

[5] M Bellare, P Rogaway. Random oracles are practical A paradigm for designing efficientprotocols. In ACM CCS, 93, ACM Press, New York . 1993; 62-73.

[6] Jacques Stern, David Pointcheval, John Malone-Lee, Nigel P Smart. Flaws in Applying Proof Methodologies to Signature Schemes . In CRYPTO 2002 LNCS 2442,Springer, Berlin. 2002; 93–110.

[7] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, Brent Waters. Sequential Aggregate Signatures and Multisignatures Without Random Oracles. In EUROCRYPT, LNCS 4004, Springer, Berlin. 2006; 465–485.

[8] D Boneh, C Gentry, B Lynn, H Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Proceedings of Euro-crypt, LNCS 2656, Springer, Berlin. 2003; 416–32.

[9] A Boldyreva. Threshold signature, multisignature and blind signature schemes based on the gap-Diffie-Hellmangroup signature scheme. In Proceedings of PKC 2003, LNCS 2567 , Springer, Berlin. 2003; 31–46.

[10] B Waters. Effcient identity-based encryption without random oracles. In Proceedings of Eurocrypt, LNCS 3494 , Springer, Berlin. 2005; 14–27.

[11] S Goldwasser, S Micali, R Rivest. A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Computing. 1988; 17(2): 281–308.

[12] M Naor, R Ostrovsky, R Venkatesan, M Yung. Perfect Zero-Knowledge Arguments for NP Using AnyOne-Way Permutation. J. Cryptology. 2008; 11(2): 87-108.

[13] I Haitner, O Horvitz, J Katz, CY Koo, R Morselli, R Shaltiel. Reducing Complexity Assumptions for Statistically-Hiding Commitment. In Proc. EUROCRYPT. 2005; 58-77.

[14] I Haitner, O Reingold. Statistically-Hiding Commitment from Any One-Way Function.In 39th STOC. 2007; 1-10.

[15] M Nguyen, SJ Ong, S Vadhan. Statistical Zero-Knowledge Arguments for NP from Any One-Way Function. In Proc. 2006; 66-76.

[16] M Nguyen, S Vadhan. Zero-Knowledge with Efcient Provers. In Proc. 38th STOC. 2006; 287-295

[17] D Catalano, I Visconti. Hybrid Commitments and Their Applications to Zero-knowledge Proof Systems. Theoretical Computer Science. 2007; 37: 229-260.

[18] O Goldreich. Foundations of Cryptography (Basic Tools), Cambridge University Press. 2001; 68-88.

[19] R Pass. Alternative Variants of Zero-Knowledge Proofs. Licentiate Thesis, Stockholm, Sweden. 2004; 121-138.

[20] CM Tang, DY Pei, ZA Yao. Etcient Zaps and Signatures of Knowledges. In Proceeding of IEEE International Conference on Computational Intelligence and Security. 2007; 637-641.

[21] CM Tang, DY Pei, ZJ Liu, XF Wang. Delegateable Signature Based on Non-interactive Witness Indistinguishable and Non-interactive Witness Hiding Proofs. Science in China Series F: Information Sciences. 2008; 68-78.

[22] CY Lin, TC Wu, F Zhang. A Structured Multisignature Scheme from the Gap Difffie-Hellman Group. Cryptology ePrint Archive, http://eprint.iacr. org/. Report 2003; 090.

[23] I Haitner, JJ Hoch, O Reingold, G Segev. Finding Collisions in Interactive Proto-cols - A Tight Lower Bound on the Round Complexity of Statistically-Hiding Commitments. *http://eprint.iacr.org/2007/145*. 2008.

[24] I Damgard. On s-protocols. CPT. Available at http://www.daimi.au.dk/ ivan/Sigma.ps. 2004