Controllable Ring Signatures and Its Application to E-Prosecution

Wei Gao
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Department of Mathematics and Informatics, Ludong University, Yantai 264025, China
Email: sdgaowei@gmail.com
Guilin Wang
School of Computer Science & Software Engineering, University of Wollongong , Wollongong NSW 2522, Australia
Email: guilin@uow.edu.au
Xueli Wang*
School of Mathematics, South China Normal University, Guangzhou 510631, China
*Corresponding Author Email: wangxuyuyan@gmail.com
Dongqing Xie
School of Computer Science & Education Software, Guangzhou University , Guangzhou 510006, China
Email: dqxie@hnu.cn

***Abstract*—This paper introduces a new concept called controllable ring signature which is ring signature with additional properties as follow. (1) Anonymous identification: by an anonymous identification protocol, the real signer can anonymously prove his authorship of the ring signature to the verifier. And this proof is non-transferable. (2) Linkable signature: the real signer can generate an anonymous signature such that every one can verify whether both this anonymous signature and the ring signature are generated by the same anonymous signer. (3) Convertibility: the real signer can convert a ring signature into an ordinary signature by revealing the secret information about the ring signature. These additional properties can fully ensure the interests of the real signer. Especially, compared with a standard ring signature, a controllable ring signature is more suitable for the classic application of leaking secrets. We construct a controllable ring signature scheme which is provably secure according to the formal definition. As an application, we design a E-prosecution scheme based on this controllable ring signature scheme and show its security.**

***Index Terms*—Certificateless cryptography; certificateless threshold decryption; provable security; random oracle model; bilinear pairing**

I. Introduction

The concept of ring signatures was introduced by Rivest, Shamir and Tauman in [2]. It enables any individual to spontaneously conscript arbitrarily $n-1$ entities and generate a publicly verifiable 1-out-of-n signature on behalf of the whole group (called a ring), yet the actual signer remains anonymous. Many extensions of a standard ring signature, such as linkable ring signature [3], convertible ring signature [4], separable ring signature [5], [6], threshold ring signature [7], ID-based ring signature [8], proxy ring signature [9], ring authenticated encryption [10], conditionally anonymous ringsignature [11] have been proposed in the literature. Ring signature and its variants have been used in many applications such as leaking secrets [2], designated verifier signature [2], anonymous identification/authentication for ad hoc groups [7], e-voting [3], e-cash, attestation in [12], bidder-anonymous english auction [13] and so on.

For the motivation of our new concept, we revisit the classic application of ring signatures in leaking secrets. Suppose that Bob (also known as "Deep Throat") is a member of the cabinet of Lower Kryptonia, and that Bob wishes to leak a juicy fact to a journalist about the escapades of the Prime Minister, in such a way that Bob remains anonymous, yet such that the journalist is convinced that the leak was indeed from a cabinet member. At a glance, it seems that a standard ring signature can help Bob to perfectly complete this task: he signs the message using a ring signature scheme on behalf of the whole cabinet. However, the following cases will show that a standard ring signature is not enough for leaking secrets in the real world.

(1) Suppose that another cabinet member Charlie is a good friend of the Prime Minister. To help the Prime Minister, Charlie generates a ring signature on an announcement. It states that he is the leaker and the previous published story about the Prime Minister is not true but a political joke. Of course, Bob's ring signature and Charlie's ring signature use the same "ring" – the whole cabinet. Now, how can Bob prevent this impersonation?
(2) Suppose that the journalist is very interested in these leaked secrets and wants to communicate with the real signer in order to discuss more details. So the journalist publishes his telephone number and wants the real signerto contact him through an anonymous

Corresponding Author: Xueli Wang, Email: wangxuyuyan @ gmail .com. This paper is the extended version of the paper [1]: Wei Gao, Guilin Wang, Xueli Wang, Dongqing Xie: Controllable Ring Signatures. WISA 2006: 1-14. This work is partially supported by National Natural Science Foundation of China (No. 60973135, No.60970111, NO.61202475) and Humanities and Social Science Research Project of the Ministry of Education (11YJCZH039).

doi:10.4304/jcp.8.4.833-841

phone call. How can Bob convince the journalist that the anonymous call is from the real signer through a untransferable proof?

(3) Suppose that Bob needs to publish further proofs for the escapades of the Prime Minister. How can Bob make people believe that both the previous secrets and these further proofs are leaked by the same anonymous cabinet member?

(4) After the disgraced Prime Minister is disposed, Bob maybe wants to remove the anonymity of the ring signature. In other words, how can Bob convert the ring signature into a standard digital signature?

Roughly speaking, (2) motivate the topic of secure anonymous identification; (3) can be captured by the notion of the linkability of anonymous signatures; (4) can be formalized as the notion of convertibility of a ring signature.

A. Related Work

Some extensions of a standard ring signature can only partially solve the above mentioned problems. In fact, the above problems were not so comprehensively pointed out in existing literature. Now we briefly review these related work.

Linkable ring signatures proposed in [3] have some limitations for leaking secrets. First, the schemes in [3] are not unconditionally but computationally anonymous. Secondly, every one can deny a ring signature if he is not the real signer. Thirdly, the real signer can't deny the ring signature generated by himself. In fact, in [3], the linkability of a ring signature was proposed mainly for restricting the real signer. For example, a linkable ring signature can prevent a ring member from generating two ring signatures on the message in the applications such as E-cash and E-voting. On the contrary, in the application of leaking secrets, the attention should be focused on how to fully ensure the interests of the real signer.

The convertible ring signature scheme proposed in [4] is the extension of a ring signature scheme proposed in [2]. It deals with only the convertibility of the ring signature scheme. And their construction cannot be trivially extended to deal with the linkability and anonymous identification. Additionally, the authors did not formalize the security model for the convertibility of ring signatures and their analysis is too simple.

The modified ring signature in [2] can guarantee only the computational anonymity. The proposed way can be used to show that a non-signer is not the real signer. A similar way can be used to show who is the real signer. In fact, they proposed a way to convert a ring signature to an ordinary signature. However, it seems difficult to extend their way to deal with the properties of linkability and anonymous authorship of a ring signature.

B. Contributions

Our contributions are twofold, as listed below. On the one hand, we revisit the classic application of ring signatures in leaking secrets and point out a list of practical problems unsolved by a standard ring signature. Motivated by these problems, we formalize the new notion of controllable ring signature. It is a useful cryptographic primitive which can fully ensure the interests of the real signer and rightly restrict him as follows.

(1) The real signer remains unconditionally anonymous unless he himself exposes his identity.

(2) Despite the unconditional anonymity, the real signer has enough power to control his signature in the sense that he can anonymously prove his authorship, generate a linkable signature, and convert the controllable ring signature.

(3) Despite the full power to control his signature, the real signer is rightly restricted since he is not able to generate a controllable ring signature and then convince a third party that it is generated by others.

(4) Despite the unconditional anonymity, any other party (non-signer) cannot abuse the anonymity. For example, there is no way for him to present the proof that the ring signature is (or not) due to him.

On the other hand, we propose an efficient construction of a controllable ring signature, which is based on the standard ring signature of Abe et al. [6]. And the underlying paradigm may also be used to transform other standard ring signatures to controllable ones.

At last, as an application, we design an E-Prosection scheme and analyze its security.

II. Framework and Security Requirements

A. Syntax of Controllable Ring Signature

*Definition 1 (*Syntax of CRS): A controllable ring signature scheme contains eight algorithms (or protocols): GenKey, RSign/RVerify, AIdentify, SSign/SVerify, Convert/CVerify as follows:

- GenKey: On input a security parameter 1^κ, it outputs a private key sk and a public key pk.
- Rsign: It takes a message m, the list, say L, of public keys $\{pk_i\}_{i=0}^{i=n-1}$ of ring members $\{A_i\}_{i=0}^{i=n-1}$ and the real signer A_k's secret key sk_k, and outputs a controllable ring signature σ and a secret information π. σ is public and π is secretly stored by A_k. We will call $\{pk_i\}_{i=0}^{i=n-1}$ or $\{A_i\}_{i=0}^{i=n-1}$ the ring for σ indiscriminatingly. And we will call a party not being A_k a non-signer. If a party is in $\{A_i\}_{i=0}^{i=n-1}$, he will be called a ring member. And a party not in $\{A_i\}_{i=0}^{i=n-1}$ will be called a non-ring-member.
- RVerify: It takes the message m, the ring L, and the controllable ring signature σ, and outputs either 1 or 0 meaning whether σ is valid for m and L or not.
- AIdentify: It is a protocol between the signer A_k and a verifier. The common inputs are the message m, the ring $\{pk_i\}_{i=0}^{i=n-1}$ and the controllable ring signature σ for m and L generated by A_k. It allows A_k to anonymously prove his authorship of σ. We require that the verifier cannot get any information about identity of the real signer from the properties of the communication channel.

- SSign: It takes m', π, σ, and outputs an anonymous signature σ' on the message m'. Here, π is the secret information associated with the controllable ring signature σ. We call σ' a linkable signature for σ.
- SVerify: It takes a message m', a controllable signature signature σ and a linkable signature σ', and outputs 1 or 0 meaning whether σ' and σ are linkable (i.e., whether σ and σ' are generated by the same anonymous ring member).
- Convert/CVerify: After the real signer of a controllable signature σ reveals the relative secret information π and his identity A_k, every one can verify whether σ is generated by A_k.

B. Security Requirements of Controllable Ring Signatures

We now describe four security requirements of a controllable ring signature scheme, which are perfect anonymity, uncontrollability, I-unforgeability, and II-unforgeability. In the following definitions, adversaries will be allowed to query some oracles: (1) A controllable ring signing oracle O_R which returns a controllable ring signature with respect to the queried message m, the ring L; (2) a converted ring signing oracle O_{CR} which returns a converted ring signature with respect to the queried message m, the ring L and the real signer A_k; (3) an anonymously identifying oracle O_A which returns an interactive proof for knowing the secret value associated with the queried controllable ring signature; (4) a linkable signing oracle O_S which returns a linkable signature on the queried message for the given controllable ring signature; (5) the corrupting oracle O_K which returns the secret key corresponding to the queried public key pk.

*Definition 2 (*Signer Anonymity): Let $L = \{pk_0, pk_1, \ldots, pk_{n-1}\}$ where each key is generated as $(pk_i, sk_i) \leftarrow \mathsf{GenKey}(1^{\kappa_i})$. A controllable ring signature scheme is perfectly signer-anonymous if, for any L, any message m, and any σ generated by $\mathsf{RSign}(m, L, sk)$ where sk is uniformly chosen from $\{sk_0, sk_1, \ldots, sk_n\}$, given (L, m, σ), any unbound adversary $\mathcal{A}^{O_A, O_S}(L, m, \sigma)$ outputs i such that $sk = sk_i$ with probability exactly $1/|L|$.

The above property ensures that the real signer remains unconditionally anonymous even after he generates linkable signatures or anonymously proves his authorship, as long as he does not convert this controllable ring signature.

*Definition 3 (*Uncontrollability against Non-Signers): Let L be the ring $\{pk_0, pk_1, \ldots, pk_{n-1}\}$ where $(pk_i, sk_i) \leftarrow \mathsf{GenKey}(1^{\kappa_i})$. Let $\kappa = \min(\kappa_0, \ldots, \kappa_{n-1})$. A controllable ring signature scheme is uncontrollable if, for any L, any message m, and any σ generated by $\mathsf{RSign}(m, L, sk)$ where $sk \xleftarrow{R} \{sk_0, sk_1, \ldots, sk_n\}$, any polynomial-time oracle machine $\mathcal{A}^{O_A, O_S}$ succeeds only with negligible probability in κ for any one of the following tasks: for the ring signature (L, m, σ) which is not converted, he tries to generate a valid linkable signature for (L, m, σ), or prove the authorship, or output(π', pk') such that $\mathsf{CVerify}(L, m, \sigma, \pi', pk') = 1$; for the converted ring signature (L, m, σ, pk, π), he tries to output another pair (π'', pk'') for $pk'' \neq pk$ s.t. $\mathsf{CVerify}(L, m, \sigma, \pi'', pk'') = 1$.

The above property ensures that a controllable ring signature cannot be controlled by any non-signer: before the controllable ring signature is converted, any non-signer cannot anonymously claim the authorship, generate a linkable signature or convert it. Furthermore, it ensures that any non-signer cannot dishonestly convert a controllable ring signature even he attains the correct converted ring signature.

Definition 4: (I-Unforgeability against Non-Ring-Members) Let (pk_i, sk_i) is generated by running $\mathsf{GenKey}(1^{\kappa_i})$ for $i = 0, \ldots, n-1$. Let κ=min$\{\kappa_0, \ldots, \kappa_{n-1}\}$ and$\mathcal{L} = \{pk_0, \ldots, pk_{n-1}\}$. A controllable ring signature scheme is existentiallyI-unforgeable against adaptive chosen-message and chosen public key attacks if, for any polynomial-time oracle machine $\mathcal{A}^{O_R}$ such that $(L, m, \sigma) \leftarrow \mathcal{A}^{O_R}(\mathcal{L})$, its output satisfies $\mathsf{RVerify}(L, m, \sigma) = 1$ only with negligible probability in κ. Restriction is that $L \subseteq \mathcal{L}$ and (L, m, σ) does not appear in the set of oracle queries and replies between $\mathcal{A}$ and O_R.

Roughly speaking, as in a standard ring signature scheme, any controllable ring signature cannot be forged by any non-ring member. Note that the above definition is almost the same to the unforgeability defined in [6] with trivial and negligible differences.

Definition 5: (II-Unforgeability of Converted Ring Signatures) Let $\mathcal{L} = \{pk_0, pk_1, \ldots, pk_{n-1}\}$ where each key is generated as $(pk_i, sk_i) \leftarrow \mathsf{GenKey}(1^{\kappa_i})$. A controllable ring signature scheme is II-unforgeable against non-signers if, any polynomial time adversary $\mathcal{A}^{O_{CR}, O_K}(\mathcal{L})$ outputs (m, L, σ, π, pk) such that $\mathsf{CVerify}(L, m, \sigma, pk, \pi) = 1$ with only negligible probability in κ. Restriction is that $\mathcal{A}$ does not get the secret key sk corresponding to pk from the oracle O_K and A does not get the converted ring signature (σ, π) with respect to (L, m, pk) from the oracle O_{CR}.

The above property ensures that: for a ring L, even if the attacker corrupts all ring members but the single one A_k which he will attack, he can not forge the converted ring signature due to the party A_k. Trivially, this property implies that the real signer is not able to dishonestly convert a ring signature into that due to the other ring member.

III. Building Blocks and the Paradigm

In this section, we briefly describe some cryptographic schemes that will be used to construct our controllable ring signature.

A. Abe et al.'s Ring Signature Scheme

Genkey': Let p_i, q_i be large primes. Let $\langle g_i \rangle$ denote a prime subgroup of $\mathbb{Z}_{p_i}$ generated by g_i whose order is q_i. Choose a random $x_i \in \mathbb{Z}_{q_i}$ as the secret key and set $y_i = g_i^{x_i} \mod p_i$. Let $H_i : \{0,1\}^* \rightarrow \mathbb{Z}_{q_i}$ be publicly available hash functions. Let $pk_i = (p_i, q_i, g_i, y_i, H_i)$ be

the DL public key of the ring member A_i. Let L be the set $\{pk_i\}_{i=0}^{n-1}$.

RSign': A_k generates a ring signature for the message m and the ring L as follows.

1) Initialization Select $\alpha \in_R \mathbb{Z}_{q_k}$ and compute $e_k = g_k^{\alpha} \mod p_k$. Compute $c_{k+1} = H_{k+1}(L, m, e_k)$.
2) Forward Sequence: For $i = k+1, \ldots, n-1, 0, \ldots, k-1$, select $s_i \xleftarrow{R} \mathbb{Z}_{q_i}$ and compute $c_{i+1} = H_{i+1}(L, m, g_i^{s_i} y_i^{c_i} \mod p_i)$.
3) Forming the ring: Compute $s_k = \alpha - c_k x_k \mod q_k$.

The resulting signature is

$$\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_0, \ldots, pk_{n-1}).$$

RVerify': A ring signature σ= $(c_0, s_0, \ldots, s_{n-1}; pk_0, \ldots, pk_{n-1})$ for the message m is verified as follows. For $i = 0, \ldots, n-1$, compute $e_i = g_i^{s_i} y_i^{c_i} \mod p_i$ and then compute $c_{i+1} = H_{i+1}(L, m, e_i)$ if $i \neq n-1$. Accept if $c_0 = H_0(L, m, e_{n-1})$. Reject otherwise.

B. Pedersen's Commitment Scheme

Pedersen's commitment scheme [14] is as follows. Let the DL public key (p, q, g, y) be generated as in the the above scheme and the secret key $\log_g y$ be generated by a trusted center. The committer commits himself to an $c \in \mathbb{Z}_q$ by choosing $s \in_R \mathbb{Z}_q$ at random and computing $E(c, s) = g^c y^s \mod p$. For $E(c, s) = g^c y^s \mod p$, $\log_g y$ is the trapdoor: given c, s and $\log_g y$, it is easy to compute another pair (c', s') such that $g^c y^s = g^{c'} y^{s'} \mod p$.

For this commitment scheme, we have the following properties (1) statistical hiding: $E(c, s)$ reveals no information about c; (2) computational binding: the committer cannot open a commitment to c as $c' \neq c$ unless he can find $\log_g y$; (3) trapdoor exposure: (c, s) and (c', s') satisfying $E(c, s) = E(c', s')$ and $(c, s) \neq (c', s')$ can be used to compute the trapdoor $\log_g y$.

There is an honest-verifier zero-knowledge protocol for proof of knowledge of the opening (c, s) for a commitment $E(c, s)$ [15]. Based on this basic protocol, it is easy to modularly construct a digital signature using the Fiat-Shamir technique [16] or to a zero-knowledge proof of knowledge of (c, s) secure against cheating verifiers using the paradigm proposed in [17].

C. A New Variant Schnorr Signature Scheme

In this section, we will construct a special digital signature scheme by sequentially applying two modular transformations [16], [18] to the well-known Schnorr identification protocol [19]. It is obvious that the resulting signature scheme is inferior to the Schnorr signature scheme, but we claim that the purpose to propose the following scheme is not for a practical digital signature scheme but for showing the security of our proposed controllable ring signature.

Now, we present the new variant Schnorr signature scheme as follows.

1) Key Generation: The signer's public key is a DL public key $pk_1 = (p_1, q_1, g_1, y_1, H_1)$ as in the above ring signature scheme. And the signing secret key is $x_1 = \log_{g_1} y_1$. Additionally, the DL public key $pk_t = (p_t, q_t, g_t, y_t, H_t)$ for the trapdoor commitment is also needed. Here, it is required that the secret key is not known by any one. In practice, such pk_t can be generated as follows.
   Let p_t and q_t be two large primes such that $q_t | p_t - 1$ and $q_t^2 \nmid p_t - 1$ and g_t be the generater of the q-order subgroup. $H_t : \{0,1\}^* \to \mathbb{Z}_{q_t}$ is the cryptographic hash function. Additionally, we also need another public hash function $H_t' : \{0,1\}^* \to \mathbb{Z}_{p_t}$. Set $y_t = H_t'(l)^{(p_t-1)/q_t} \mod p_t$ where l can be any publicly known string, e,g., $l = p_t || q_t || g_t$.
   Note that if $q_t | p_t - 1$ and $q_t^2 \nmid p_t - 1$, then $r^{\frac{p_t-1}{q_t}} \mod p_t$ is always an element generated by g_t for any $r \in \mathbb{Z}_{p_t}^*$. Also note that it is easy to check whether p_t, q_t, g_t, y_t (with public l) are honestly generated. And given honestly generated p_t, q_t, g_t, y_t, it is infeasible for one to get $\log_{g_t} y_t$. For simplicity, we just assume that p_t, q_t, g_t, y_t, H_t are public parameters where $\log_{g_t} y_t$ is not known by anyone.
2) Signing: Given the message m, first select $\alpha \in_R \mathbb{Z}_q$ and compute $e = g_1^{\alpha} \mod p_1$. Then compute the Pedersen's commitment of e as $e' = g_t^{H_t(e)} y_t^r \mod p_2$ where $r \in_R \mathbb{Z}_{q_t}$. Next, compute $c = H_1(m, e')$ and $s = \alpha - c x_1 \mod q_1$. The output signature $\sigma = (c, s, r)$.
3) Verification: Given the signature $\sigma = (c, s, r)$ and the message m, check whether

$$c = H_1(m, g_t^{H_t(g_1^s y_1^c \mod p_1)} y_t^r \mod p_t).$$

We give the security analysis as follows. First, we review the two underlying paradigms for the above scheme. In [18], the Damgård's paradigm was proposed to modularly turn a special honest-verifier zero-knowledge protocol (called Σ-protocol) into a concurrent zero-knowledge proof of knowledge in the auxiliary string model (i.e., it is assumed that the secret key for the trapdoor commitment is not known by any one except the trusted party). The Fiat-Shamir paradigm [16] is widely used to modularly construct a digital signature scheme secure in the random oracle model from a three-pass secure identification against passive attacks [20]. It is easy to see that the above scheme is constructed by sequentially applying the Damgård's transformation and the Fiat-Shamir paradigm to the Schnorr identification protocol. The unforgeability of the digital signature can be modularly derived from the properties of the two paradigms [18], [20]. Here, we omit the straightforward and lengthy security proof from scratch. In more details, we have the following lemma which will be used to show the security of the controllable ring signature scheme:

Claim 1: If the hash function H_1 is assumed to be a random oracle, the other hash function H_2 is collision-resistant, the secret key $\log_{g_t} y_t$ for the commitment is not be known by anyone and the discrete logarithm problem is intractable, then the above digital signature scheme is existentially unforgeable against adaptively chosen-message attacks.

IV. Proposed Controllable Ring Signature Scheme

A. Paradigm for Constructing Controllable Ring Signatures

Note that for an ordinary ring signature, although every ring member can anonymously generate a signature, he has to "close the ring" at his own position using his own secret key. If the real signer hides some proof for the "closing position" in the ring signature (in our construction, we perfectly hide the proof through Pedersen's commitment scheme.), he will be able to control it as follows. On the one hand, before the hidden proof is public, this controllable ring signature is just like a standard ring signature. And the real signer can anonymously prove his authorship, or generate linkable ring signatures by using the hidden proof as the secret key. After the hidden proof is public, this controllable ring signature is converted into a standard signature generated by the real signer.

B. The Proposed Scheme

Our scheme is the extension of the above reviewed ring signature scheme from [6] as follows.

Genkey: A user's key (pk, sk) of the DL-type is generated as in Genkey′. Additionally, the DL public key $pk_t = (p_t, q_t, g_t, y_t, H_t)$ for the trapdoor commitment is also needed. Here, it is required that the secret key is not known by any one. It can be generated as described in the new variant Schnorr digital signature scheme in Section 3.3.

RSign/RVerify: A signer A_k generates a controllable ring signature for the message m and the ring L, in the following way.

1) Initialization (1) Select $\alpha \in_R \mathbb{Z}_{q_k}$ and compute $e_k = g_k^{\alpha} \mod p_k$. (2) Compute $c_t = H_t(e_k)$, select $s_t \in_R \mathbb{Z}_{q_t}$, and then compute $e_t = g_t^{c_t} y_t^{s_t} \mod p_t$. (3) Compute $c_{k+1} = H_{k+1}(L, m, e_k, e_t)$.
2) Forward Sequence: For $i = k+1, \ldots, n-1, 0, \ldots, k-1$, select $s_i \xleftarrow{R} \mathbb{Z}_{q_i}$ and compute $e_i = g_i^{s_i} y_i^{c_i} \mod p_i$ and set $c_{i+1} = H_{i+1}(L, m, e_i, e_t)$.
3) Forming the ring: Compute $s_k = \alpha - c_k x_k \mod q_k$.

The resulting ring signature is

$$\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$$

and the real signer will store the secret information (c_t, s_t).

A controllable ring signature

$$\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$$

for the message m is verified as follows. For $i = 0, \ldots, n-1$, compute $e_i = g_i^{s_i} y_i^{c_i} \mod p_i$ and then compute $c_{i+1} = H_{i+1}(L, m, e_i, e_t)$ if $i \neq n-1$. Accept if $c_0 = H_0(L, m, e_{n-1}, e_t)$. Reject otherwise.

Note that we refer the reader to the 3 facts in the next section for the basic idea underlying the above construction and the next protocols or algorithms.

AIdentify: For a valid controllable ring signature $\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$ of the message m, the real signer anonymously proves his authorship of σ through a zero-knowledge proof of knowledge of (c_t, s_t) s.t. $e_t = g_t^{c_t} y_t^{s_t} \mod p_t$ as follows:

1) The verifier randomly chooses c', s', t_1', t_2' and computes $e' = g_t^{c'} y_t^{s'} \mod p_t$, $x' = g_t^{t_1'} y_t^{t_2'} \mod p_t$. Then (e', x') is sent to the prover.
2) The real signer picks random numbers $t_1, t_2 \in \mathbb{Z}_{q_t}^*$, and computes $x = g_t^{t_1} y_t^{t_2} \mod p_t$. Then the real signer randomly selects $r_1'', r_2'', z'' \in \mathbb{Z}_{q_t}$ and computes $x'' = g_t^{r_1''} y_t^{r_2''} e'^{z''} \mod p_t$. Next the real signer randomly selects $z' \in \mathbb{Z}_{q_t}$. At last, (x, x'', z') is sent to the verifier.
3) The verifier computes $r_1' = t_1' - z'c' \mod q_t, r_2' = t_2' - z's' \mod q_t$, choose a random number $\tilde{z} \in \mathbb{Z}_{q_t}$ and sends $(r_1', r_2', \tilde{z})$ to the real signer.
4) First, the real signer checks whether $x' = g_t^{r_1'} y_t^{r_2'} e'^{z'} \mod p_t$. If so, the real signer sends to the verifier z'', r_1'', r_2'' and (z, r_1, r_2) such that:
$$z = z'' \oplus \tilde{z},\ r_1 = t_1 - zc_t, r_2 = t_2 - zs_t.$$
5) The verifier will accept that the prover is the real signer of σ if $x = g_t^{r_1} y_t^{r_2} e_t^z \mod p_t$, $x'' = g_t^{r_1''} y_t^{r_2''} e'^{z''} \mod p_t$ and $\tilde{z} = z'' \oplus z$. Otherwise, he will reject it.

Here note that, as in Def.3, we implicitly assume that the verifier has obtained the authentic ring signature before he requires the anonymous proof. In fact, this can be easily implemented. For example, he can sign the ring signature using his secret key, sends it to the real signer and requires anonymous proof for the authorship of this signed ring signature.

SSign/SVerify: For a valid controllable ring signature $\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t;\ pk_0, \ldots, pk_{n-1})$ on the message m, the linkable signature (z, r_1, r_2) on a message m' is generated as follows:

$$t_1, t_2 \xleftarrow{R} \mathbb{Z}_{q_t}^*, x = g_t^{t_1} y_t^{t_2} \mod p_t, z = H_t(m', x),$$
$$r_1 = t_1 - zc_t \mod q_t, r_2 = t_2 - zs_t \mod q_t.$$

The verifier will accept that (z, r_1, r_2) and σ is signed by the same anonymous signer if $H_t(m', g_t^{r_1} y_t^{r_2} e_t^z \mod p_t) = z$ and reject otherwise.

Convert/CVerify: To convert a controllable ring signature σ, the real signer A_k releases the relative s_t such that $e_t =$

$g_t^{H_t(g_k^{s_k} y_k^{c_k} \mod p_k)} y_t^{s_t} \mod p_t$. (σ, s_t) will be called the converted ring signature due to the party A_k.

To check whether (σ, r) is a valid converted ring signature due to the party A_k, the verifier checks whether σ is a valid controllable ring signature through RVerify and checks whether

$$e_t = g_t^{H_t(g_k^{s_k} y_k^{c_k} \mod p_k)} y_t^{s_t} \mod p_t$$

where c_k is computed as in RVerify.

Remark 1: In the above scheme, given a controllable ring signature, there is no way for the receiver to check whether this ring signature can be correctly converted. In other words, for a controllable ring signature, the verifier can only check whether it is generated by a ring member but can not check whether it is controllable. However, in some applications, it may be necessary for the verifier to be convinced of the convertibility. In fact, the above scheme can be easily extended to support a non-interactive proof for the convertibility of the controllable ring signature. We will show that the proof for controllability can be implemented using 1-out-of-n witness indistinguishable proofs with a concrete discrete logarithm setting [21].

Concretely speaking, to convince the receiver of the controllability, the real signer should present an non-interactive proof of knowledge of (c_t, s_t) such that:

$$e_t = g_t^{c_t} y_t^{s_t} \mod p_t,$$
$$c_t \in \{H_t(e_0), H_t(e_1), \ldots, H_t(e_{n-1})\}$$

where $e_i = g_i^{s_i} y_i^{c_i} \mod p_i$ for $i = 0, \ldots, n-1$. The above proof is equivalent to the proof knowledge of s_t such that

$$e_t g_t^{-c_t} = y_t^{s_t} \mod p_t,$$
$$c_t \in \{H_t(e_0), H_t(e_1), \ldots, H_t(e_{n-1})\}.$$

In other words, the real signer should prove knowledge of one of the n logarithms $\log_{y_t}(e_t g_t^{-H_t(e_0)}), \ldots, \log_{y_t}(e_t g_t^{-H_t(e_{n-1})})$. According to [21], this kind of non-interactive proof of 1-out-of-n knowledge in a concrete discrete logarithm setting can be easily constructed.

V. Security Analysis

Before analyzing the security of the above controllable ring signature, we first point the following simple facts about the basic tools in our scheme without detailed explanation:

Fact 1. RSign/RVerify *is same to the ring signature (all discrete case) proposed in [6] except that e_t is inserted in our controllable ring signature.*

Fact 2. Aldentify *is a zero-knowledge proof of knowledge of* (c_t, s_t) satisfying $e_t = g_t^{c_t} y_t^{s_t} \mod p_t$.

Sketch of proof: This protocol is modularly constructed by applying the paradigm proposed in [17] to the honest-verifier zero-knowledge proof of knowledge of the opening of the Pedersen's commitment [15]. In more details, the verifier first present the commitment e' of the value t_1 and then proves the knowledge of the opening. Next, the prover proves that he knows the opening of e' or e_t. The fact that Adentify is zero-knowledge proof of knowledge of (c_t, s_t) can be modularly derived from the paradigm [17]. Here we omit the proof from scratch.

Fact 3. SSign /SVerify *is transformed from the identification protocol based DLP (Here the public key is $e_t = g_t^{c_t} y_t^{s_t}$ and (c_t, s_t) is the secret key)due to Okamoto [15] via the Fiat-Shamir technique [16].*

Based on the above facts, we can easily analyze the security of our proposed controllable ring signature informally.

Theorem 1: The above scheme is unconditionally anonymous.

Proof (1). From the probabilistic process of RSign, we can see that: (a) all $s_i, 0 \le i \le n-1$, are randomly distributed in $\mathbb{Z}_{q_i}$; (b) e_t is randomly distributed in $\mathbb{Z}_{p_t}$ since $s_t \in_R \mathbb{Z}_{q_t}$ and e_k is randomly distributed in $\mathbb{Z}_{p_k}$; (c) c_0 is also fixed when $L = \{pk_i\}_{i=1}^n, m, e_t, e_k, s_0, \ldots, s_{n-1}$ are fixed. So for fixed L, m, the distribution of $(e_t, c_0, s_1, \ldots, s_{n-1})$ is independent of the public key of the real signer.

(2). First, the protocol Aldentify is zero-knowledge secure against cheating verifiers. Especially, the proof is witness-indistinguishable since the proof is independent of which of $\{(c_t, s_t) | e_t = g_t^{c_t} y_t^{s_t} \mod p_t\}$ used by the prover. Second, the linkable signature (z, r_1, r_2) is determined by the random chosen (t_1, t_2) and independent of which of $\{(c_t, s_t) | e_t = g_t^{c_t} y_t^{s_t} \mod p_t\}$ used by the signer. So there is no information of (c_t, s_t) leaked through the protocol Aldentify and the linkable signatures.

Combining (1) and (2), we can see that for a controllable ring signature, the ring signature itself, the anonymous proof of authorship and the linkable signatures are all independent of which of (c_t, s_t) in $\{(c_t, s_t) | e_t = g_t^{c_t} y_t^{s_t} \mod p_t\}$. So we can conclude that the identity of the real signer is unconditionally protected as long as the real signer does not exposes his identity to the verifier.

Theorem 2: The above scheme is uncontrollable.

Proof Let $\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$ be a controllable ring signature where $e_t = g_t^{c_t} y_t^{s_t} \mod p_t$.

From the Fact 2,3, it is obvious that the attacker can control a controllable ring signature through any of Aldentify, SSgin, Convert only if he know (c_t, s_t) s.t. $e_t = g_t^{c_t} y_t^{s_t} \mod p_t$. However, before the real signer publishes (s_t, c_t), c_t is unconditionally hidden in e_t. And the attacker cannot get (c_t, s_t) by accessing the oracle corresponding to Aldnetify since Aldentify is zero-knowledge. Neither can the attacker get (c_t, s_t) by querying the O_S oracle because of Fact 3. So before (c_t, s_t) is public, no non-signer can control the controllable ring signature.

According to CVerify, if (σ, s_t) and (σ, s_t') are valid converted ring signatures due to A_k and $A_{k'}$ respectively, then we have $e_t = g_t^{c_t} y_t^{s_t} = g_t^{c_t'} y_t^{s_t'} \mod p_t$, where

$c_t = H_t(g_k^{s_k} y_k^{c_k} \mod p_k), c'_t = H_t(g_{k'}^{s_{k'}} y_{k'}^{c_{k'}} \mod p_{k'})$. By two different opening of the same e_t, the trapdoor $\log_{g_t} y_t$ can be easily derived. However, in our scheme, it is infeasible for one to compute $\log_{g_t} y_t$. So after a controllable ring signature σ is converted, any non-signer cannot prove that σ was not generated by A_k.

Theorem 3: In the random oracle model, our controllable ring signature scheme is I-unforgeable against non-ring-members if Abe et al.'s ring signature is existentially unforgeable against adaptive chosen-message and public key attacks.

Proof After comparing the definitions of the I-unforgeability and the unforgeability in [6], and the two ring signing algorithms of RSign in Section 3.1 and RSign′ in Section 4, it is straightforward to derive the conclusion.

Theorem 4: Our controllable ring signature scheme is II-unforgeable if the signature scheme in Section 3.3 is existentially unforgeable against adaptively chosen-message attacks.

Proof For the formal definition of existential unforgeability against adaptively chosen-message attacks, we refer the readers to [22]. Let $\mathcal{F}_1$ be the II-forger attacking our controllable ring signature scheme. We will use it to construct a (adaptively chosen-message attacker) forger $\mathcal{F}_2$ attacking the signature scheme in Section 3.3. The challenger for $\mathcal{F}_2$ provides the signing public key $\overline{pk}$, the committing public key $pk_t = (p_t, q_t, g_t, y_t, H_t)$ and the signing oracle which returns a valid signature on the queried message.

First, $\mathcal{F}_2$ simulates the ring $\mathcal{L}$ in which one is the the public key $\overline{pk}$ and the others are generated by himself using Genkey. Here note that for the public keys generated by himself, $\mathcal{F}_2$ knows the secret keys. $\mathcal{F}_2$ initialize $\mathcal{F}_1$ by sending the ring $\mathcal{L}$ and the public key $pk_t = (p_t, q_t, g_t, y_t, H_t)$. Second, when $\mathcal{F}_1$ queries the signing oracle O_{CR} on the message m, the ring $L = \{pk_0, pk_1, \ldots, pk_{|L|-1}\} \subset \mathcal{L}$, and the public key $pk_k \in L$, $\mathcal{F}_2$ will simulates the converted ring signature due to pk_k as follows. If $pk_k \neq \overline{pk}$, with the secret key sk_k relative to pk_k, $\mathcal{F}_2$ uses RSign and Convert to generate a converted ring signature and returns it. If $pk_k = \overline{pk}$, $\mathcal{F}_2$ queries its challenger on the message $m' = (L, m, g_{k-1}^{\alpha_{k-1}})$ where $\alpha_{k-1} \in_R \mathbb{Z}_{q_{k-1}}$. After receiving the signature (c_k, s_k, r), $\mathcal{F}_2$ computes $e_k = g_k^{s_k} y_k^{c_k} \mod p_k$ and $e_t = g_t^{H_t(e_k)} y_t^r \mod p_t$, and sets $c_{k+1} = H_{k+1}(L, m, e_k, e_t)$. Then, for $i = k+1, \ldots, |L|-1, 0, \ldots, k-2$, $\mathcal{F}_2$ selects $s_i \in_R \mathbb{Z}_{q_i}$ and computes $c_{i+1} = H_{i+1}(L, m, g_i^{s_i} y_i^{c_i}, e_t)$. For $i = k-1$, compute $s_{k-1} = \alpha_{k-1} - c_{k-1} x_{k-1} \mod q_{k-1}$. Now $\mathcal{F}_2$ returns the converted signature (σ, r) where $\sigma = (c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$. It is obvious that the converted ring signatured (σ, r) is valid only if (c_k, s_k, r) is a valid signature.

Third, when $\mathcal{F}_1$ queries the corrupting oracle O_K on the public key in $\mathcal{L}$, $\mathcal{F}_2$ returns the secret key if this public key is generated by $\mathcal{F}_2$. Otherwise, $\mathcal{F}_2$ aborts.

At last, $\mathcal{F}_1$ returns a converted ring signature (σ, r) due to pk_k on the message m, the ring $L \subset \mathcal{L}$. Let σ be $(c_0, s_0, \ldots, s_{n-1}; pk_t, e_t; pk_0, \ldots, pk_{n-1})$. If $pk_k = \overline{pk}$, then $\mathcal{F}_2$ returns (c_k, s_k, r) as the signature on the message $m' = (L, m, e_{k-1})$. If $pk_k \neq \overline{pk}$, $\mathcal{F}_2$ aborts. Here, it is obvious that $c_k = H(L, m, e_{k-1}, g_t^{H_t(g_k^{s_k} y_k^{c_k} \mod p_k)} y_t^r \mod p_t)$ if (σ, r) is valid converted ring signature due to pk_k.

Now, we analyze the probability that $\mathcal{F}_2$ does not aborts. Note that in the above simulation, all the public keys in the $\mathcal{L}$ play the same roles and $\overline{pk}$ cannot be distinguished from the other public keys. Since at least one public key in the $\mathcal{L}$ is not corrupted, so the probability that the public key $\overline{pk}$ is not queried on the oracle O_K is at least $\frac{1}{|\mathcal{L}|}$. The probability that $\mathcal{F}_1$ returns the converted ring signature corresponding to $\overline{pk}$ is at least $\frac{1}{|\mathcal{L}|}$. So The probability that $\mathcal{F}_2$ does not aborts is at least $\frac{1}{|\mathcal{L}|^2}$. Since a valid converted ring signature (σ, r) implies that $c_k = H_k(L, m, e_{k-1}, g_t^{H_t(g_k^{s_k} y_k^{c_k} \mod p_k)} y_t^{s_t} \mod p_t)$, c_k, s_k, r is just a digital signature with respect to the signature scheme in Section 3.3 with the public key pk_k and the message $m' = (L, m, e_{k-1})$. So if $\mathcal{F}_1$ can succeed in forging a valid converted ring signature with probability larger than ϵ_1, then $\mathcal{F}_2$ succeeds in attacking the digital signature scheme in Section 3.3 with probability $\epsilon_2 \geq \frac{1}{|\mathcal{L}|^2}\epsilon_1$. By Lemma 1, the II-unforgeability of our controllable ring signature is obtained.

VI. E-Prosecution Scheme Based on Controllable Signatures

In this section, based on the above controllable signature scheme, we design the E-prosecution scheme as follows. This E-prosecution scheme involves two parties: the public authority such as the police office, and the group (ring) of all possible prosecutors. By this scheme, the prosecutor can prosecute sequential messages First and i-th offline prosection, and even anonymously initiates an online discussion with the authority (Online Anonymous Prosection), and collect the reward by opening this identity to authority (Award Collection). As will be shown in the security analysis, this E-Prosecution can well protect the identity privacy of the prosecutor.

- System Setup: In this phase, the public authority A_t and the possible prosecutors A_i $(0 \leq in-1)$ generates their public/privat key pairs respectively. Just like Genkey, for each possible prosecutor A_i indexed by i, let p_i, q_i be large primes. Let $\langle g_i \rangle$ denote a prime subgroup of $\mathbb{Z}_{p_i}$ generated by g_i whose order is q_i. Choose a random $x_i \in \mathbb{Z}_{q_i}$ as the secret key and set $y_i = g_i^{x_i} \mod p_i$. Let $H_i : \{0,1\}^* \to \mathbb{Z}_{q_i}$ be publicly available hash functions. Let $pk_i = (p_i, q_i, g_i, y_i, H_i)$ be the DL public key of the ring member A_i. Let L be the set $\{pk_i\}_{i=0}^{n-1}$. Similarly, the public authority A_t

generates his public key $pk_i = (p_t, q_t, g_t, y_t, H_t)$ and the private key x_t such that $y_t = g_t^{x_t}$.

- First Offline Anonymous Prosecution: In this phase, the real prosecutor A_k decides the first prosecution message m_1, generates the ring signature $\sigma^1 = (c_0^1, s_0^1, \ldots, s_{n-1}^1; pk_t^1, e_t^1; pk_0^1, \ldots, pk_{n-1}^1)$, by running the algorithm RSign. Then the real prosecutor sends (σ^1, m_1) to the authority. The authority can check whether this prosecution comes from one of the ring by running the algorithm RSign.
- i-th Offline Anonymous Prosecution ($i > 1$): Here, our prosecution scheme can provide the real prosecutor the ability to continue prosecuting some messages. In this way, the receiver can check whether these sequential prosecuted messages come from the original prosecutor, although the real prosecutor remains anonymous. In this phase, to prosecute the i-th message m_i, the real prosecutor A_k generate the signature $\sigma_i = (z^i, r_1^i, r_2^i)$ by running the algorithm SSign. Then he sends $(\sigma^1, m_1, \sigma^i, m_i)$ to the authority. By running RVerify,SVerify, the authority can check whether these two signature were generated by the same prosecutor which is anonymous in the ring.
- Online Anonymous Prosecution. In some cases, the online anonymous discussion between the real prosecutor A_k and the authority A_t may be needed. For example, required by the prosecutor or the authority, the real prosecutor may call the the authority for discussing some details on his prosecution. To this end, the prosecutor can anonymously authenticate himself by running the protocol AIdentify. Once the anonymous authentication is accepted, the authority can assure the real prosecutor anonymously and can discuss some details on the prosecution with the prosecutor.
- Award Collection. At last, after running Convert, the real prosecutor A_k can prove that he is the real prosecutor by showing s_t such that $e_t = g_t^{H_t(g_k^{s_k} y_k^{c_k} \mod p_k)} y_t^{s_t} \mod p_t$. By running CVerfity, the authority can check this fact. If the authority accepts, the real prosecutor will get his reward.

Next, we analyze the security of the above E-Prosecution scheme.

- The prosecutor remains anonymous even after he opens this identity to the authority.
  Before opening the commitment $e_t = g_t^{c_t} y_t^{s_t}$, the identity relative information $c_t = H_t(e_k)$ is unconditionally secure in the information theory sense. This is because for every possible value of $c_t = H_t(e_{k'})(k' \neq k)$ corresponding to any possible prosecutor in the ring, there exists a value s_t such that $e_t = g_t^{c_t} y_t^{s_t}$.
  After opening (c_t, s_t) to authority, the authority can generate one value s_t for any value $c_t = H_t(e_{k'})$ $(k' \neq k)$. Since he can arbitrarily open e_t for any possible prosecutor in the ring, any third party will not believe the authority's opening. Hence, the prosecutor always remains secure.
- After the first offline prosecution and before award collection, only the real prosecutor knows the opening (c_t, s_t) for the commitment $e_t = g_t^{c_t} y_t^{s_t}$, according to the discrete logarithm assumption. Here note that before the real prosecutor discloses his identity to the authority, even the authority himself can not open the commitment. In fact, directly opening e_t still means solving the discrete logarithm for the authority. Of course, after the prosecutor open the commitment to the authority, he can arbitrarily open the commitment using his secret key as the trapdoor. Hence, only the real prosector can make sequential offline prosection, anonymous online prosection and award collection.
- If the prosecutor wants to further make the prosecuted message secret, he can (1) generate all the relative signatures on the commitment $m' = g_t^m y_t^r$ instead of the message m, (2) encrypt the message m into the ciphertext c using a certain public key encryption scheme with the y_t as the public key, (3) and sends the relative signature σ, the partial opening value r and the ciphertext c to the prosecutor. In this way, firstly, any third party can not obtain the message m from the communication procession. Secondly, the authority can not prove to one third party that there is one prosecutor who prosecuted the message m. This because the authority can use his private key as the commitment trapdoor to arbitrarily open the commitment $m' = g_t^m y_t^r$ for any possible message m.
- The real prosecutor cannot frame any other party in the ring. The reason is that after the prosecution, if the real prosecutor wants to maliciously claim that it is not himself but a certain other party $A_{k'}$ who made the prosecution, he will face the problem of working out a new opening s_t' for $e_t = g_t^{c_t'} y_t^{s_t'}$ where $c_t' = H_t(g_{k'}^{s_{k'}} y_{k'}^{c_{k'}} \mod p_{k'})$. Without the trapdoor x_t such that $y_t = g_t^{x_t}$, this operation is infeasible for the real prosecutor.

VII. Conclusion

In this paper, we revisited the classic application of a ring signature in leaking secrets and point out a list of problems unsolved by a standard ring signature. Motivated these problems, we formalized a new cryptographic concept called a controllable ring signature and propose a concrete scheme. This extension of a standard ring signature can fully ensure the interests of the real signer: (1) the real signer remains unconditional anonymous as long as he does not remove anonymity; (2) only the real signer can control the ring signature: only he can anonymously prove the authorship, generate a linkable ring signature or convert it. On the other hand, a ring member is rightly restricted since he can not generate a controllable ring signature and convince one that it is generated by others. At last, using this controllable

ring signature scheme, we design a secure E-prosecution scheme.

Acknowledgement. We would like to express our gratitude thanks to the anonymous referees of WISA 2006 for their invaluable suggestions to improve this paper.

REFERENCES

[1] Wei Gao, Guilin Wang, Xueli Wang, Dongqing Xie: Controllable Ring Signatures. WISA 2006: 1-14.
[2] A. Shamir, R. Rivest and Y. Tauman, How to leak secret. Asiacrypt'01, LNCS 2248, pp.552-565. Springer-Verlag, 2001.
[3] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). ACISP'04, LNCS 3108, pp.325-335. Springer-Verlag, 2004.
[4] K.C. Lee, H.A. Wen, and T. Hwang. Convertible ring signature. IEE Proc.-Commun., Vol. 152, No. 4, pp.411- 414, August 2005.
[5] J. K. Liu, Victor K. Wei, and Duncan S. Wong. A separable threshold ring signature scheme. ICISC 2003, LNCS 2971, pp.12-26. Springer-Verlag, 2003.
[6] M. Abe, M. Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. Asiacrypt 2002, LNCS 2501, pp.415-432. Springer-Verlag, 2002.
[7] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to Ad-hoc groups. Crypto 2002, LNCS 2442, pp.465-480. Springer-Verlag, 2002.
[8] Sherman S.M. Chow, Richard W.C. Lui, Lucas C.K. Hui, and S.M. Yiu. Identity based ring signature: why, how and what next. EuroPKI 2005, LNCS 3545, pp.144-161. Springer-Verlag, 2005.
[9] Yong Yu, Bo Yang, Fagen Li, Mingwu Zhang, An efficient proxy ring signature scheme, Journal of Beijing University of Posts and Telecommunications, Vol.30, No.3, pp.23-27, 2007.
[10] Jiqiang Lv, Kui Ren, Xiaofeng Chen, Kwangjo Kim: The ring authenticated encryption scheme - How to provide a clue wisely. Inf. Sci. 179(1-2): 161-168 (2009)
[11] Shengke Zeng, Shaoquan Jiang, Zhiguang Qin. An efficient conditionally anonymous ringsignature in the random oracle model. Theoretical Computer Science, to appear, http://dx.doi.org/10.1016/j.tcs.2012.01.027.
[12] Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. Information Security Practice and Experience (ISPEC 2005), LNCS 3439, pp.48-60. Springer-Verlag, 2005.
[13] Hu Xiong, Zhong Chen, Fagen Li: Bidder-anonymous English auction protocol based on revocable ring signature. Expert Syst. Appl. 39(8): 7062-7066 (2012)
[14] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. Crypto'91. LNCS 576, pp.129-149. Springer-Verlag, 1991.
[15] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes,Crypto'92, LNCS 740, pp.31-53. Springer-Verlag, 1993.
[16] A.Fiat and A.Shamir. How to prove yourself: Practical solutions to identification and signature problems. Crypto'86, LNCS 263, pp.186-199. Springer-Verlag, 1986.
[17] R. Cramer, I. Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. PKC 2000, LNCS 1751, pp.354-372. Springer-Verlag, 1986.
[18] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. Eurocrypt 2000, LNCS 1807, pp.418-430. Springer-Verlag, 2000.
[19] C. P. Schnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161-174, 1991.
[20] M. Abdalla, J. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: minimizing assumptions for security and forward-security. Eurocrypt 2002, LNCS 2332, pp.418-433. Springer-Verlag, 2002.
[21] R.Cramer, I.Damgård, and B.Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. Crypto'94, LNCS 839, pp.174-187. Springer-Verlag,1994.
[22] S.Goldwasser, S.Micali, and R.Rivest. A digital signature scheme secure against adaptive chosen message attacks. SIAM Journal of Computing, 17(2):281-308, 1988.

Wei Gao received his Ph.D., MS and BS degrees in applied mathematics from Hunan University in 2006, Guangzhou University 2003, and Ludong University in 2000, respectively. He is a lecturer in Ludong University from 2007. From 2010, he is Posdoc at Shanghai Jiaotong University. His research interests include security, cryptography and number theory.

Guilin Wang received her Ph.D degree in computer science, from Institute of Software, Chinese Academy of Sciences, P. R. China, in 2001. He is currently a senior lecturer in University of Wollongong, Australia. His research interests include cryptography and information security.

Xueli Wang received his PhD degree in mathematics from the Academy of China in 1991, his MS degree in mathematics from Shannxi Normal University in 1987. He is currently Professor of Computer Science at South China Normal University. His current research interests include cryptography, number theory and elliptic curves.

Dongqing Xie received his PhD degree in mathematics from Hunan University in 1999, his MS degree in computer science from Xidian University in 1988. He is currently Professor of Computer Science at Guangzhou University. His current research interests include applied cryptography and information security.