Linkable Ring Signatures from Linear Feedback Shift Register*

Dong Zheng[1,3], Xiangxue Li[2], Kefei Chen[1], Jianhua Li[2]

[1] Department of Computer Science and Engineering, Shanghai JiaoTong University
dzheng@sjtu.edu.cn
[2] School of Information Security Engineering, Shanghai JiaoTong University
[3] National Laboratory for Modern Communications, Chengdu

Abstract. Linkable ring signatures can simultaneously provide the properties of anonymity, spontaneity as well as linkability. Linear feedback shift register (LFSR) sequence can be used to shorten the representation of elements in a field. This paper proposes an LFSR-based linkable ring signature scheme, whose main computation operations are performed in *base field* $GF(q)$ whereas security properties are under the state based discrete logarithm assumption(S-DLA)(and a new state based computational assumption weaker than state based decisional Diffie-Hellman assumption). The latter potentially says that the scheme is secure in the *extension field* $GF(q^d)$(d the stage of the LFSR). All these make our scheme a flexible primitive for ubiquitous computing in which information processing has been thoroughly integrated into everyday objects and activities.

Keywords. Characteristic sequence, Linear feedback shift register, Ring signatures, Anonymity, Linkabilty.

1 Introduction

For many practical applications or resource-limited environments, it is often desirable to speed up the cryptosystems without notable security degradation. Recently, several cryptosystems have been proposed to shorten the representation of the elements in the finite field[3, 7, 11, 13] by representing them with the coefficients of their minimal polynomials. For instance, Niederreiter[11] designed encryption and key agreement schemes based on general n-th order linear feedback shift register (LFSR) sequences. Giuliani and Gong[3] proposed a general class of LFSR-based key agreement and signature schemes based on n-th order characteristic sequences. Main contributions in these work are that they do not require as much bandwidth as their counterparts based on finite fields.

Ring signature attracts significant attention since its invention[1, 12, 15, 16]. In a ring signature scheme, a user first selects a set U (called a ring) of possible signers including himself, then signs a message using his private key and

* Supported by NSFC(No.60573030, 60673076, 60672068) and NCET(No.NCET-06-0393).

the public keys of all the members in the ring. The resulting signature can be verified to be generated by some user in the ring, but the identity of the actual signer will not be revealed, hence the signature provides the signer the property of anonymity which cannot be revoked. Linkable ring signatures have a specific property of linkability which means that any one can tell if two ring signatures are generated by using the same private key. In other words, linkability says two signatures by the same actual signer can be identified as such, but the signer remains anonymous. The first linkable ring signature scheme was proposed by Liu *et.al.* [10]. The property of linkability is really essential in some scenarios as explained below: (1) Suppose there is an organization that wants to conduct an anonymous and voluntary questionnaires among its members. It is demanded that only legitimate members can submit the questionnaires, and at the same time, each member cannot submit more than one questionnaire. Conventional ring signatures can ensure those who submitted the questionnaires are members of the organization and maintain users' anonymity, but they cannot prevent a member from submitting more than one questionnaire. (2) Another practical instance is to detect double-voting in an e-voting system. Although blind signatures or other cryptographic protocols seem to be able to achieve this goal, yet they require all the users to participate in the *setup* stage even they do not intend to join subsequent protocols.

In current work, we will construct a linkable ring signature scheme based on d-th characteristic sequences generated by an LFSR. Main computation operations of the scheme are performed in the base field $GF(q)$. In fact, besides hash evaluations and addition/multiplications in Z_P, only multiplications of elements in $GF(q)$ are involved in our scheme. This particularly produces a fast system as no exponentiation in $GF(q^d)$ is required. As for its security properties, by resorting to the random oracle methodology, we can show that it is secure under the state based discrete logarithm assumption and state based decisional product Diffie-Hellman assumption as defined in Section 2. Since state based discrete logarithm problem is proved to be equivalent to traditional DLP in $GF(q^d)$ [3], the proposed scheme successfully enhances the security of the system, at the same time, with low computational costs. In other words, to get a system equivalent to one based on extension field $GF(q^d)$, there is no need to compute any exponentiation in $GF(q^d)$.

Organization. The rest of the paper is organized as follows. We first introduce some conceptions and notations related to d-th characteristic sequences in Section 2, then give a security model of linkable ring signatures in Section 3. Section 4 is devoted to the new ring signature scheme based on d-th characteristic sequences. Its formal security arguments are described in Section 5. Finally, concluding remarks are made in Section 6.

Notations. Throughout this paper, let Z_P denote the set $\{0, 1, 2, \cdots, P-1\}$, and Z_P^* denote $Z_P \backslash \{0\}$. By $\in_R S$, it means choosing a random element from the set S with a uniform distribution. For an algorithm $\mathcal{A}$, we use $x \leftarrow \mathcal{A}$ to denote that $\mathcal{A}$ is executed on some specified input and its output is assigned to the variable x; if $\mathcal{A}$ is a probabilistic algorithm, we write $x \overset{R}{\leftarrow} \mathcal{A}$. Finally,

throughout this paper, we often equate a user with his identity, his public key or his secret key without risks of confusion according to the context.
Negligible Function. We say a function $f : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for every constant $c \geq 0$, there exists an integer k_c such that $f(k) < k^{-c}$ for all $k > k_c$.

2 Preliminaries

2.1 LFSR sequences

We briefly review the necessary about linear feedback shift register. Let q be a prime or a power of prime, $f(x) = x^d + a_1 x^{d-1} + a_2 x^{d-2} + \ldots + a_d$ $(a_i \in GF(q), i = 1, ..., d)$ be an irreducible polynomial over $GF(q)$ with a root α of order P in the extension field $GF(q^d)$. A sequence $s = \{s_j\}$ over $GF(q)$ is said to be an LFSR sequence generated by $f(x)$ if $s_{j+d} = a_1 s_{j+d-1} + a_2 s_{j+d-2} + \ldots + a_d s_j$ $(j \geq 0)$.

If an initial state of s is given by $s_j = tr(\alpha^j), j = 0, 1, \ldots, d-1$, where $tr(\cdot)$ is the trace map from $GF(q^d)$ to $GF(q)$, then s is called a d-th order characteristic sequence. It is well-known that the period of the d-th characteristic sequence s is equal to the order P of α. Thus we can define $s_j = s_{P+j}$ for all $j \leq 0$, and further consider the sequence $\{s_j\}$ with indices running over all integers. We denote the i-th state of the LFSR sequence as $\bar{s}_i = (s_i, s_{i+1}, \ldots, s_{i+d-1})$, and set $A_j = (s_j, s_{2j}, \ldots, s_{rj})$, where r is defined by

$$r = \begin{cases} d-1 & \text{for general } q \text{ and } d, \\ d/2 & \text{if } q = p^2, \text{and } d \text{ is even,} \\ (d-1)/2 & \text{if } q = p^2 \text{ and } d \text{ is odd.} \end{cases}$$

Vector A_j can be used to recover the minimal polynomial of γ^j, $(\gamma \in GF(q^d), j \in \mathbb{Z})$[3]. Refer to [4] for more details about the theory of LFSR sequences.

2.2 Complexity problems

We start this part with several main sequence operations, *i.e.*, SO1, SO2 and SO3, which will be repetitively employed in our scheme. Both SO1 and SO2 can be performed efficiently by the existing algorithms[3], and SO3 can be viewed to be derived from SO1 and SO2[8]. The following sequence operations can be jointly used to design *smart* and *efficient* cryptographic primitives, including our construction as depicted in Section 3.

- Sequence Operation 1(SO1): *Given A_j and an integer l($0 < j, l < P$), to compute A_{jl}.*
- Sequence Operation 2(SO2): *Given states $\bar{s}_j$ and $\bar{s}_l$($0 < j, l < P$), to compute $\bar{s}_{j+l}$.*
- Sequence Operation 3(SO3): *Given $\bar{s}_j$ and an integer l($0 < j, l < P$), to compute $\bar{s}_{jl}$.*

We proceed to recall the definitions of state based discrete logarithm problem(S-DLP)(and state based decisional Diffie-Hellman problem(S-DDHP)) on which the securities of our scheme are based.

Definition 1. *The problem **S-DLP** is, given $(q, n, P, \bar{s}_1, \bar{s}_j)$, to compute j. For a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, we define his **advantage** against the S-DLP as*

$$Adv_{\mathcal{A}}^{\text{S-DLP}} \overset{\text{def}}{=} \Pr\left[\mathcal{A}(\bar{s}_1, \bar{s}_j) = j\right],$$

where the probability is taken over the random coins consumed by $\mathcal{A}$.

*We say that the (t, ϵ)-**S-DL assumption(S-DLA)** holds, if no t-time adversary $\mathcal{A}$ has advantage at least ϵ in solving the S-DLP.*

Definition 2. *The problem **S-DDHP** is, given $(q, n, P, \bar{s}_1, \bar{s}_u, \bar{s}_v, \bar{s}_w)$, to decide wether $w = uv$ holds. For a PPT adversary $\mathcal{A}$, we define his **advantage** against the S-DDHP as*

$$Adv_{\mathcal{A}}^{\text{S-DDHP}} \overset{\text{def}}{=} |\Pr\left[\mathcal{A}(\bar{s}_1, \bar{s}_u, \bar{s}_v, \bar{s}_{uv}) = 1\right] - \Pr\left[\mathcal{A}(\bar{s}_1, \bar{s}_u, \bar{s}_v, \bar{s}_w) = 1\right],$$

where the probability is taken over the random coins consumed by $\mathcal{A}$.

*We say that the (t, ϵ)-**S-DDH assumption** holds, if no t-time adversary $\mathcal{A}$ has advantage at least ϵ in solving the S-DDHP.*

It is known that the state based discrete logarithm problem as defined above is computationally equivalent to the traditional DLP in $GF(q^d)$[14], and that the complexity of breaking S-DDH assumption is equivalent to that of solving decisional Diffie-Hellman problem in the field $GF(q^d)$[3]. Generally speaking, computational problems such as the DLP are much harder than the DDH, *i.e.*, $DDH \leq DLP$[6]. Analogical claims come into existence in the state based scenarios[3, 14]. In the following, we further introduce a new problem called *state based decisional product Diffie-Hellman (S-DPDH) problem.* Resorting to the new problem(and S-DLP), we can present our linkable ring signature scheme and construct formal security arguments for the scheme.

Definition 3. *The state based decisional product Diffie-Hellman(**S-DPDH**) problem is, given $(q, d, P, \bar{s}_1, \bar{s}_a, \bar{s}_j, \bar{s}_l, \bar{s}_{ac})$, to decide whether $c = jl$ holds. More concretely, for a PPT adversary $\mathcal{A}$, we define his **advantage** against the problem S-DPDH as*

$$Adv_{\mathcal{A}}^{\text{S-DPDH}} \overset{\text{def}}{=} |\Pr[\mathcal{A}(\bar{s}_1, \bar{s}_a, \bar{s}_j, \bar{s}_l, \bar{s}_{ajl}) = 1 - \Pr[\mathcal{A}(\bar{s}_1, \bar{s}_a, \bar{s}_j, \bar{s}_l, \bar{s}_{ac}) = 1]|,$$

where the probability is taken over the random coins consumed by $\mathcal{A}$.

*S-DPDH assumption says that S-DPDH problem is hard to solve. More precisely, we say that the (t, ϵ)-**S-DPDH assumption(S-DPDHA)** holds, if no t-time adversary $\mathcal{A}$ has advantage at least ϵ in solving the S-DPDH problem.*

One can easily note that S-DDHP is just an instance of S-DPDH problem(when we fix $\bar{s}_a = \bar{s}_1$). In other words, S-DPDHA is *no stronger than* S-DDH assumption, which makes the problem S-DPDH independently interesting. It is believed that we prefer to build cryptographic systems on weaker assumptions. In fact, our construction is based on the two weak assumptions S-DLA and S-DPDHA which make the proposed scheme more flexible.

3 Framework of Linkable Ring Signatures

In this section, we will describe the definitions of linkable ring signatures ($\mathcal{LRS}$) and of the security notions for $\mathcal{LRS}$.

3.1 Linkable Ring Signatures

We first give an overview for the $\mathcal{LRS}$ model. On the one hand, as original ring signatures, $\mathcal{LRS}$ contains the system initialization algorithm Setup, user key generation algorithm KeyGen, signature generation algorithm Sign and signature verification algorithm Verify. On the other hand, $\mathcal{LRS}$ has a special algorithm called Link from which any verifier can decide whether two given ring signatures are generated by using the same secret key.

Definition 4. *A linkable ring signature scheme $\mathcal{LRS}$ consists of a tuple of five polynomial-time algorithms:*

Setup: *a probabilistic algorithm, taking as input the security parameter 1^λ, returns a public common parameter param. We write param $\overset{R}{\leftarrow}$ Setup(λ).*

KeyGen: *a probabilistic key generation algorithm, taking as input the system parameter param and a user's identity $ID \in \{0,1\}^*$, returns the public/secret key pair (PK, SK) for the user. We write $(PK, SK) \overset{R}{\leftarrow}$ KeyGen(param, ID);*

Sign: *a probabilistic signing algorithm, taking as input the system parameter param, a message m, the secret key SK_i of the actual signer with identity ID_i, and the public keys $PK_1, ...,$ PK_n, returns the resulting signature σ. We write $\sigma \overset{R}{\leftarrow}$ Sign(param, SK_i, PK_1, $..., PK_n, m$);*

Verify: *a deterministic verification algorithm, taking as input the system parameter param, a candidate signature σ on the original message m and the public keys $PK_1, ..., PK_n$, returns 1 if (m, σ) is a valid signature, and 0 otherwise. We write (1 or 0) $\overset{R}{\leftarrow}$ Verify(param, $PK_1, ..., PK_n, m, \sigma$);*

Link: *The algorithm takes as inputs two valid signatures σ_1 and σ_2, and returns either 1 for* **linkable** *or 0 for* **unlinkable**. *We write (0 or 1) $\overset{R}{\leftarrow}$ Link(σ_1, σ_2).*

3.2 Security Notions for Linkable Ring Signatures

Next we will formalize the security notions for $\mathcal{LRS}$. To this end, we consider the following oracles which together model the abilities of an adversary against $\mathcal{LRS}$:

– $\mathcal{H}(\cdot)$: a *random oracle* is a theoretical black box that responds to every query with a (truly) random response chosen uniformly from its output domain, except that for any specific query, it responds the same way every time it receives that query. Put another way, a random oracle is a mathematical function mapping every possible query to a random response from its output domain.

- $\mathcal{CO}(\cdot)$: a corruption oracle, upon receiving an identity $ID_i \in \{0,1\}^*$, returns the corresponding secret key SK_i;
- $\mathcal{SO}(\cdot,\cdot)$: a signing oracle, taking as input a set of users L and a message m, outputs a signature of L;

For the security for $\mathcal{LRS}$, there are three aspects we should consider: anonymity, unforgeability and linability.

Let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link})$ be an $\mathcal{LRS}$ scheme, n a polynomial and $\mathcal{A}$ an adversary attacking the property of anonymity for DGS. Our model takes insider attack into account by allowing the adversary to corrupt some fraction of the members and thereby come into possession of their secret keys. $\mathcal{A}$ runs in three stages.

In the find stage the adversary is given an initial information string I and the public keys of the members in the ring. It outputs two identities, say, ID_0, ID_1 for uncorrupted members and a message $m \in \{0,1\}^*$. Based on a challenge bit b, one of the two identities is selected to yield a challenge signature on the message m, which is returned to the adversary, now in its guess stage. Finally $\mathcal{A}$ returns a bit d as its guess of the challenge bit b. In each stage the adversary will output state information that is returned to it in the next stage. We now provide a formal definition.

Definition 5. *Let* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link})$ *be an* $\mathcal{LRS}$ *scheme. For a PPT adversary* $\mathcal{A}$*, let* n *be a polynomial,* $b \in \{0,1\}$*, consider the experiment:*

```
Experiment $\mathbf{Exp}^{\mathbf{IA\text{-}b}}_{\Pi,\mathcal{A}}(k)$
    $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
    For $i = 0, ..., n-1$ do $(x_i, y_i) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
    $(ID_0, ID_1; m; st) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot),\mathcal{CO}(\cdot),\mathcal{H}(\cdot)}(find,\ I, y_0, ..., y_{n-1})$;
    $\sigma \overset{R}{\leftarrow} \mathsf{Sign}(I, x_b, y_0, ..., y_{n-1}, m)$;
    $d \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot),\mathcal{CO}(\cdot),\mathcal{H}(\cdot)}(guess,\ \sigma; st)$;
    return $d$.
```

Herein, we naturally require that $\mathcal{A}$ *did not submit* ID_0, ID_1 *to the corruption oracle* $\mathcal{CO}(\cdot)$*. The advantage of the adversary is defined as*

$$Adv^{\text{IA}}_{\mathcal{A},\Pi}(k) \overset{\text{def}}{=} \Pr[\text{Exp}^{\text{IA-0}}_{\Pi,\mathcal{A}}(k) = 0] - \Pr[\text{Exp}^{\text{IA-1}}_{\Pi,\mathcal{A}}(k) = 0]$$

Let Π=$(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be an* $\mathcal{LRS}$ *scheme. We say that it is irrevocably anonymous if the function* $Adv^{\text{IA}}_{\mathcal{A},\Pi}(k)$ *is negligible for any poly(k)-time adversary* $\mathcal{A}$ *and any polynomial* n*.*

Definition 6. *Let* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link})$ *be an* $\mathcal{LRS}$ *scheme. For a PPT adversary* $\mathcal{A}$ *whose goal is to forge a ring signature, let* n *be a polynomial, consider the experiment:*

```
Experiment $\mathbf{Exp}^{\mathbf{UF}}_{\Pi,\mathcal{A}}(k)$
    $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
    For $i = 0, ..., n-1$ do $(x_i, y_i) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
    $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot),\mathcal{H}(\cdot)}(I, y_0, ..., y_{n-1})$.
    $\mathcal{A}$ wins if $\mathsf{Verify}(param,\ m^*,\ \sigma^*,\ y_0,\ \cdots,\ y_{n-1}) = 1$.
```

Above, It is mandated that m^* was not queried to the signing oracle.

The advantage of the adversary is defined as $Adv^{\mathrm{UF}}_{\mathcal{A},\Pi}(k) \overset{\mathrm{def}}{=} \Pr[\mathcal{A} \ \ \text{wins}]$. Let Π=(Setup, KeyGen, Sign, Verify, Link) be an $\mathcal{LRS}$ scheme. We say that it is unforgeable if the function $Adv^{\mathrm{UF}}_{\mathcal{A},\Pi}(k)$ is negligible for any poly(k)-time adversary $\mathcal{A}$ and any polynomial n.

The notion of linkability allows anyone to determine whether two signatures have been issued by the same member in the ring. For simplicity, we say an $\mathcal{LRS}$ scheme is linkable if no member in the group can generate two signatures σ_1, σ_2 such that $\mathsf{Link}(\sigma_1, \sigma_2)$ returns 0. More formally, we have

Definition 7. *Let Π = (Setup, KeyGen, Sign, Verify, Link) be an $\mathcal{LRS}$ scheme. For a PPT adversary $\mathcal{A}$ whose goal is to break the property of linkability, let n be a polynomial, consider the experiment:*

```
Experiment $\mathbf{Exp}^{\mathbf{FL}}_{\Pi,\mathcal{A}}(k)$
    $I \overset{R}{\leftarrow} \mathsf{Setup}(k)$;
    For $i = 0, ..., n-1$ do $(x_i, y_i) \overset{R}{\leftarrow} \mathsf{KeyGen}(I)$ EndFor;
    $\tau \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot),\mathcal{H}(\cdot)}(I, y_0, ..., y_{n-1})$ where $\tau \in \{0, ..., n-1\}$.
    $(m_1, \sigma_1; m_2, \sigma_2) \leftarrow \mathcal{A}^{\mathcal{S}(\cdot,\cdot),\mathcal{H}(\cdot)}(I, y_0, ..., y_{n-1}, x_\tau)$.
    return 1-$\mathsf{Link}(\sigma_1, \sigma_2)$.
```

It is mandated that $(m_1, \sigma_1), (m_2, \sigma_2)$ are not output of the signing oracle. The advantage of the adversary is defined as $Adv^{\mathrm{FL}}_{\mathcal{A},\Pi}(k) \overset{\mathrm{def}}{=} \Pr[\mathrm{Exp}^{FL}_{\Pi,\mathcal{A}}(k) = 1]$. Let Π=(Setup, KeyGen, Sign, Verify, Link) be an $\mathcal{LRS}$ scheme. We say that it is fully linkable if the function $Adv^{\mathrm{FL}}_{\mathcal{A},\Pi}(k)$ is negligible for any poly(k)-time adversary $\mathcal{A}$ and any polynomial n.

4 LFSR-based Linkable Ring Signatures

Previously, linear feedback shift register(LFSR) is prevalently used to generate pseudo-random sequences which are essential in stream cipher [4]. In [5], Gong and Harn studied 3-rd order LFSR sequences over a finite field whose cryptographic properties are employed to construct public-key distribution scheme and RSA-type encryption algorithm. Recently, Giulian and Gong [3] proposed an ElGamal-like LFSR-based signature scheme without formal security proof. Provable security is an important research area in cryptography. Cryptographic primitives or protocols without a rigorous proof cannot be regarded as secure in practice. There are many schemes that are originally thought as secure being successfully cryptanalyzed, which clearly indicates the need of formal security assurance. With provable security, we are confident in using cryptographic applications to replace the traditional way in physical world.

Current section is devoted to our LFSR-based linkable ring signature scheme $\mathcal{LLRS}$. As will be seen from the scheme below, one of its advantages is that its securities rely on hard problems in *extension field* $GF(q^d)$, while all computation operations are performed in *base field* $GF(q)$. This potentially speeds up the run of the scheme without notable security degradation.

More concretely, our scheme $\mathcal{LLRS}$ consists of the following five algorithms where three sequence operations SO1, SO2, SO3 are repeatedly called.

Setup: given a security parameter 1^λ, the algorithm generates the appropriate system parameter as $param = \{q,\ d,\ \bar{s}_1,\ P, H\}$ where H be a cryptographically secure hash function.

KeyGen: a user with identity ID_i randomly chooses his secret key $(w_{i,1}, w_{i,2}) \in Z_P^{*2}$, and generates matching public key $PK_i = (\bar{s}_{w_{i,1}}, \bar{s}_{w_{i,2}})$.

Sign: without loss of generality, we assume that user k is the actual signer. Let $L = \{PK_i, i = 1, \cdots, n\}$ be the collection of all the public keys, and m be the message to be signed. Figure.1. illustrates main compositions of algorithm Sign, and each box hits the high spots of each move in the algorithm.

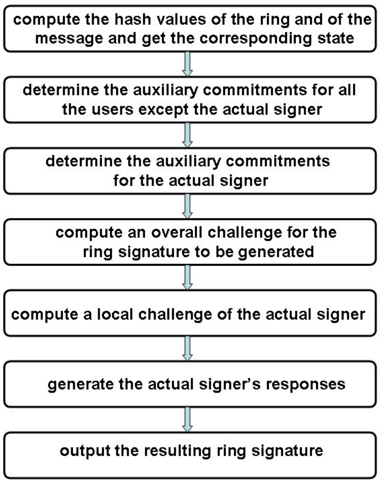
compute the hash values of the ring and of the message and get the corresponding state
determine the auxiliary commitments for all the users except the actual signer
determine the auxiliary commitments for the actual signer
compute an overall challenge for the ring signature to be generated
compute a local challenge of the actual signer
generate the actual signer's responses
output the resulting ring signature

Fig.1. Ring Signature Generation Algorithm Sign

1. Compute $h = H(L) \bmod P$, $h_1 = H(m) \bmod P$ and obtain $\bar{s}_h$ ($\bar{s}_{h_1}$, *resp.*) from $\bar{s}_1$ and $h(h_1$, *resp.*) using SO3. Set $v = hw_{k,1}w_{k,2}$ and get $\bar{s}_v$. Choose $b \in_R Z_P$, set $t_1 = hw_{i,1}$, $t_2 = h_1 b$, $t = t_1 + t_2$, and get $\bar{s}_t$.
2. For $i = 1, ..., n(i \neq k)$, perform the following steps.
   (a) Randomly pick the challenge $0 < c_i < P$, and the responses $0 < z_{i,1}, \cdots, z_{i,4} < P$.

(b) Denote $d'_{i,1}$ ($d'_{i,2}$, $d'_{i,3}$, $d''_{i,3}$, $d'''_{i,3}$, $d'_{i,4}$, $d''_{i,4}$,$d'''_{i,4}$, *resp.*) as the product $w_{i,1}c_i$ ($w_{i,2}c_i$, $hz_{i,1}$, $h_1z_{i,3}$, tc_i, $tz_{i,2}$, $h_1(-z_{i,4})$, vc_i, *resp.*), and set $d_{i,1} = z_{i,1} + d'_{i,1}$, $d_{i,2} = z_{i,2} + d'_{i,2}$, $d_{i,3} = d'_{i,3} + d''_{i,3} + d'''_{i,3}$ $d_{i,4} = d'_{i,4} + d''_{i,4} + d'''_{i,4}$.

(c) Compute $\bar{s}_{z_{i,1}}$ ($\bar{s}_{d'_{i,1}}$, $\bar{s}_{d_{i,1}}$, $\bar{s}_{z_{i,2}}$, $\bar{s}_{d'_{i,2}}$, $\bar{s}_{d_{i,2}}$, $\bar{s}_{d'_{i,3}}$, $\bar{s}_{d''_{i,3}}$, $\bar{s}_{d'''_{i,3}}$, $\bar{s}_{d'_{i,4}}$, $\bar{s}_{d''_{i,4}}$, $\bar{s}_{d'''_{i,4}}$, *resp.*) from the 2-tuple ($\bar{s}_1$, $z_{i,1}$)(($\bar{s}_{w_{i,1}}$, c_i), ($\bar{s}_{z_{i,1}}$, $\bar{s}_{d'_{i,1}}$), ($\bar{s}_1$, $z_{i,2}$), ($\bar{s}_{w_{i,2}}$, c_i), ($\bar{s}_{z_{i,2}}$, $\bar{s}_{d'_{i,2}}$), ($\bar{s}_h$, $z_{i,1}$), ($\bar{s}_{h_1}$, $z_{i,3}$), ($\bar{s}_t$, c_i), ($\bar{s}_t$, $z_{i,2}$), ($\bar{s}_{h_1}$, $-z_{i,4}$), ($\bar{s}_v$, c_i), *resp.*), and obtain $\bar{s}_{d_{i,3}}$ and $\bar{s}_{d_{i,4}}$ from the 3-tuple ($\bar{s}_{d'_{i,3}}$, $\bar{s}_{d''_{i,3}}$, $\bar{s}_{d'''_{i,3}}$) and ($\bar{s}_{d'_{i,4}}$, $\bar{s}_{d''_{i,4}}$, $\bar{s}_{d'''_{i,4}}$), respectively.

(d) States $\bar{s}_{d_{i,1}}$, $\bar{s}_{d_{i,2}}$, $\bar{s}_{d_{i,3}}$ and $\bar{s}_{d_{i,4}}$ are viewed as the auxiliary commitments for the user i.

3. Randomly pick $r_{k,1}$, $\cdots$, $r_{k,4}$, let $d_{k,1} = r_{k,1}$, $d_{k,2} = r_{k,2}$, $d'_{k,3} = hr_{k,1}$, $d''_{k,3} = h_1r_{k,3}$, $d_{k,3} = d'_{k,3} + d''_{k,3}$, $d'_{k,4} = tr_{k,2}$, $d''_{k,4} = h_1(-r_{k,4})$, $d_{k,4} = d'_{k,4} + d''_{k,4}$, and compute $\bar{s}_{d_{k,1}}$ ($\bar{s}_{d_{k,2}}$, $\bar{s}_{d'_{k,3}}$, $\bar{s}_{d''_{k,3}}$, $\bar{s}_{d_{k,3}}$, $\bar{s}_{d'_{k,4}}$, $\bar{s}_{d''_{k,4}}$, $\bar{s}_{d_{k,4}}$, *resp.*) from ($\bar{s}_1$, $d_{k,1}$) (($\bar{s}_1$, $d_{k,2}$), ($\bar{s}_h$, $r_{k,1}$), ($\bar{s}_{h_1}$, $r_{k,3}$), ($\bar{s}_{d'_{k,3}}$, $\bar{s}_{d''_{k,3}}$), ($\bar{s}_t$, $r_{k,2}$), ($\bar{s}_{h_1}$, $-r_{k,4}$), ($\bar{s}_{d'_{k,4}}$, $\bar{s}_{d''_{k,4}}$), respectively). States $\bar{s}_{d_{k,1}}$, $\bar{s}_{d_{k,2}}$, $\bar{s}_{d_{k,3}}$ and $\bar{s}_{d_{k,4}}$ are viewed as the auxiliary commitments for the actual user.
4. Set c_0 as the hash value of m, $\bar{s}_v$, $\bar{s}_t$, $\bar{s}_{d_{i,j}}$, for $1 \leq i \leq n, 1 \leq j \leq 4$. Compute $c_k = c_0 - (c_1 + ... + c_{k-1}) - (c_{k+1} + ... + c_n)$.
5. Compute the responses of user k:

$$z_{k,1} = r_{k,1} - c_k w_{k,1} \bmod P, \quad z_{k,2} = r_{k,2} - c_k w_{k,2} \bmod P,$$
$$z_{k,3} = r_{k,3} - c_k b \bmod P, \quad z_{k,4} = r_{k,4} - c_k w_{k,2} b \bmod P.$$

6. Output the signature σ as the collection of $\bar{s}_v$, $\bar{s}_t$, c_i, $z_{i,j}$, for $1 \leq i \leq n, 1 \leq j \leq 4$.

Remark 1. Although our scheme does not result in signatures with constant size, its particular properties, *i.e.*, low computational costs and high security level, make it adaptable for practical application. And of cause, it is interesting to explore an LFSR-based linkable ring signature scheme with constant size such that it speeds up the system without notable security degradation.

Verify: given a purported signature σ of a ring L on a message m, a verifier can check its validity via the following process.

1. Compute $h = H(L) \bmod P$ and $h_1 = H(m) \bmod P$, then determine $\bar{s}_h$ and $\bar{s}_{h_1}$ from ($\bar{s}_1$, h) and ($\bar{s}_1$, h_1), respectively.
2. For $i = 1, ..., n$:
   - Denote $d'_{i,1}$ ($d'_{i,2}$, $d'_{i,3}$, $d''_{i,3}$, $d'''_{i,3}$, $d'_{i,4}$, $d''_{i,4}$, $d'''_{i,4}$, *resp.*) as the product $w_{i,1}c_i$ ($w_{i,2}c_i$, $hz_{i,1}$, $h_1z_{i,3}$, tc_i, $tz_{i,2}$, $h_1(-z_{i,4})$, vc_i, *resp.*), and set $d_{i,1} = z_{i,1} + d'_{i,1}$, $d_{i,2} = z_{i,2} + d'_{i,2}$, $d_{i,3} = d'_{i,3} + d''_{i,3} + d'''_{i,3}$ $d_{i,4} = d'_{i,4} + d''_{i,4} + d'''_{i,4}$.
   - Compute $\bar{s}_{z_{i,1}}$ ($\bar{s}_{d'_{i,1}}$, $\bar{s}_{d_{i,1}}$, $\bar{s}_{z_{i,2}}$, $\bar{s}_{d'_{i,2}}$, $\bar{s}_{d_{i,2}}$, $\bar{s}_{d'_{i,3}}$, $\bar{s}_{d''_{i,3}}$, $\bar{s}_{d'''_{i,3}}$, $\bar{s}_{d'_{i,4}}$, $\bar{s}_{d''_{i,4}}$, $\bar{s}_{d'''_{i,4}}$, *resp.*) from the 2-tuple ($\bar{s}_1$, $z_{i,1}$) (($\bar{s}_{w_{i,1}}$, c_i),($\bar{s}_{z_{i,1}}$, $\bar{s}_{d'_{i,1}}$), ($\bar{s}_1$, $z_{i,2}$), ($\bar{s}_{w_{i,2}}$, c_i), ($\bar{s}_{z_{i,2}}$, $\bar{s}_{d'_{i,2}}$), ($\bar{s}_h$, $z_{i,1}$), ($\bar{s}_{h_1}$, $z_{i,3}$), ($\bar{s}_t$, c_i), ($\bar{s}_t$, $z_{i,2}$), ($\bar{s}_{h_1}$, $-z_{i,4}$), ($\bar{s}_v$, c_i), *resp.*).

- Obtain $\bar{s}_{d_{i,3}}$ and $\bar{s}_{d_{i,4}}$ from the 3-tuple ($\bar{s}_{d'_{i,3}}$, $\bar{s}_{d''_{i,3}}$, $\bar{s}_{d'''_{i,3}}$) and ($\bar{s}_{d'_{i,4}}$, $\bar{s}_{d''_{i,4}}$, $\bar{s}_{d'''_{i,4}}$), respectively. (Note that $\bar{s}_{d_{i,1}}$ and $\bar{s}_{d_{i,2}}$ are computed in step ii.)

3. Accept the signature if the hash value of m, $\bar{s}_v$, $\bar{s}_t$, $\bar{s}_{d_{i,j}}$, matches the sum of c_i, for $1 \leq i \leq n, 1 \leq j \leq 4$.

Link: taking as input two valid signatures $\sigma = (..., \bar{s}_v, ...)$ and $\sigma' = (..., \bar{s}_{v'}, ...)$, the algorithm outputs 1 (for linkable) if $\bar{s}_v = \bar{s}_{v'}$; outputs 0 (for unlinkable), otherwise.

This ends the description of our $\mathcal{LLRS}$. Consistency requires that $\forall m, m_1, m_2 \in M, j \in \{1, ..., n\}$, Verify($param$, PK_1, $\cdots$, PK_n, m, σ)=1 and Link(σ_1, σ_2) $= 1$ hold, where σ = Sign($param$, SK_j, PK_1, $\cdots$, PK_n, m), σ_i = Sign($param$, SK_j, PK_1, $\cdots$, PK_n, m_i), $i = 1, 2$ and M denotes the message space. One can easily check that the proposed scheme provides the property of consistency. Next we briefly discuss the performance as follows. In fact, above ring signature is an instantiation of the following signature proof-of-knowledge (SPK): $\sigma = SPK\{(w_1, w_2) : \bigvee_{1 \leq i \leq n}(y_{i,1} = \bar{s}_{w_1} \ \wedge \ y_{i,2} = \bar{s}_{w_2} \wedge \ \bar{s}_v = \bar{s}_{hw_1w_2})\}(m)$ which can be viewed as an extension of the following SPK(see [2] and other papers listed in [9] for further details on the notations and theories): $\sigma = SPK\{(x_1, x_2) : \bigvee_{1 \leq i \leq n}(y_{i,1} = g^{x_1} \ \wedge \ y_{i,2} = g^{x_2} \ \wedge \ v = h^{x_1 x_2})\}(m)$.

The online complexity of signature verification requires $4n$ SO2 and $10n + 2$ SO3 operations. All computations are performed by matrix multiplication in the *based field* $GF(q)$ [3, 8], which bring a quite efficient system. The advantages of $\mathcal{LLRS}$ are summarized below:

i. As will be seen in the coming section, by resorting to the random oracle methodology, its security properties are formally showed under two weak assumptions, *i.e.*, S-DLA and S-DPDHA. This is really interesting since S-DPDHA is weaker than S-DDH assumption.

ii. Main computation operations are performed in the base field $GF(q)$. In fact, besides hash evaluations and addition/multiplications in Z_P, only multiplications of elements in $GF(q)$ are involved in our scheme. This particularly produces a fast system.

Due to the fact that state based discrete logarithm problem is proved to be equivalent to traditional DLP in extension field $GF(q^d)$ [14] and that the complexity of breaking S-DDH assumption equates that of solving DDH in $GF(q^d)$ [3], the proposed scheme successfully enhances the security of the system, at the same time, with low computational costs. In other words, to get a system equivalent to one based on extension field $GF(q^d)$, there is no need to compute any exponentiation in $GF(q^d)$.

5 Security Arguments

As for the security properties of our proposed scheme, we have the following results.

Theorem 1. *Suppose $\mathcal{A}$ is a (t, ϵ)-algorithm which forges a linkable ring signature in time at most t with success probability at least ϵ, there exists an algorithm $\mathcal{B}$ which solves the S-DLP.*

Theorem 2. *The proposed scheme provides the property of signer anonymity under the S-DPDHA assumption in the random oracle model.*

Theorem 3. *The proposed scheme provides the property of linkability in the random oracle model under the assumption that S-DLP problem is hard.*

Due to space limitation, we omit the long-winded proofs which are similar to those in [17] according to the special structures of the schemes.

6 Conclusion

Linkable ring signatures can eliminate the registration phase in e-voting systems based on blind signatures and other protocols. From d-th characteristic sequences generated by an LFSR, the paper introduced a linkable ring signature scheme which can be supported by formal security arguments. The scheme enjoys the following attractive features: (i) the proposal provides an option for some applications (*e.g.* e-voting system, E-cash scheme); (ii) an independently interesting assumption(weaker than S-DDH) is introduced; (iii) main computation operations are performed in $GF(q)$; and (iv) security properties of the schemes are equivalent to those of systems based on the multiplication group $GF(q^d)$. The appealing features make our schemes more flexible and highly adaptable to practical applications in the sense that they enhance the security of the system and meanwhile maintain low computational costs as well.

In our scheme, the sizes of public keys and the resulting ring signatures are not ideal as desired. It would be very nice to find an LFSR-based signature scheme in which both the public key and the signature can be represented by short representations. Finally, it would also be nice to develop other applications dependent upon various state based problems.

References

1. M.Abe, M.Ohkubo, K.Suzuki. 1-out-of-n signatures from a variety of keys. Advances in Cryptology-Asiacrypt 2002, pp.415-432, 2002.
2. J.Camenisch, M.Stadler. Proof systems for general systems of discrete logarithms. ETH Technical Report No, 260, 1997, available at: ftp://ftp.inf.ethz.ch/pub/publications/tech-reports/.
3. K.Giuliani, G.Gong. New LFSR-Based cryptosystems and the trace discrete log problem (Trace-DLP). Proceedings of sequences and their applications-SETA 2004, pages 298-312, 2004.
4. S.Golomb. Shift register sequences. Laguna Hills, CA: Aegean Park, 1982.
5. G.Gong, L.Harn. Public-key cryptosystems based on cubic finite field extensions. IEEE Transaction on Information Theory, vol.24, pages 2601-2605, 1999.

6. N.Koblitz, A.Menezes. Another look at generic group. Cryptology ePrint Archive, 2006/230, available at: http://eprint.iacr.org/2006/230.
7. A.Lenstra, E.Verheul. The XTR public key system. Advances in Cryptology-Crypto 2000, pages 1-19, 2000.
8. X.Li, D.Zheng, K.Chen. LFSR-based signatures with message recovery. Intenational Journal of Network Security, 4(3), pages 266-270, 2007.
9. H.Lipmaa. Proofs of knowledge of certain problems. Awailable at: http://www.cs.ut.ee/ lipmaa/crypto/link/ zeroknowledge/pok.php.
10. J.Liu, V.Wei, D.Wong. Linkable spontaneous anonymous group signature for ad hoc groups. Proceedings of Australasian Conf. Information Security and Privacy-ACISP 2004, pages 325-335, 2004.
11. H.Niederreiter. Finite fields and cryptology. Proceedings of Finite fields,coding theory, and Advances in communications and computing, Dekker, New York, pages 359-373, 1992.
12. R.Rivest, A.Shamir, Y.Tauman. How to leak a secret. Advances in Cryptology-Asiacrypt 2001, pages 552-565, 2001.
13. P.Smith, C.Skinner. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms. Advances in Cryptology-Asiacrypt 1994, pages 357-364, 1994.
14. C.Tan, X.Yi, C.Siew. On the n-th order shift register based discrete logarithm. IEICE Transaction on Fundamentals, vol.E86-A(5), pages 1213-1216, 2003.
15. V.Wei. A bilinear spontaneous anonymous threshold signature for ad hoc groups. Cryptology ePrint Archive, 2004/039, available at: http://eprint.iacr.org/.
16. F.Zhang, K.Kim. ID-Based blind signature and ring signature from pairings. Advances in Cryptology-Asiacrypt 2002, pages 535-547, 2002.
17. D.Zheng, V.Wei, K.Chen. GDH group-based signature scheme with linkability. 153(5), pages 639-644, Communications, IEE Proceedings.