Knowl Inf Syst
DOI 10.1007/s10115-009-0273-4

REGULAR PAPER

Restrictive partially blind signature for resource-constrained information systems

Weidong Qiu · Zheng Gong · Bozhong Liu · Yu Long · Kefei Chen

Received: 26 January 2009 / Revised: 22 October 2009 / Accepted: 31 October 2009

Abstract Restrictive partially blind signature, which is designed for privacy-oriented information systems, allows a user to obtain a blind signature from a signer while the blind message must obey some certain rules. In order to reduce storage and communication costs, several public-key cryptosystems are constructed using characteristic sequences generated by linear feedback shift register (LFSR). In this paper, we present a new partially blind signature scheme with the restrictive property, which is based on nth order characteristic sequences generated by LFSR. By assuming the intractability of the discrete logarithm problem, our sequence-based schemes are provably secure in the random oracle model. We also present a practical e-cash application based on our restrictive partially blind signature. Due to the reduced representation of finite field elements and feasible sequence operations from LFSR, our scheme is time- and storage-efficient on both of signer and user sides. The advantages will make privacy-oriented applications more practical for resource-constrained devices.

Keywords Linear feedback shift register sequence · Partially blind signature · Restrictiveness · Electronic cash

This work is supported by National 863 Projects of China No. 2007AA01Z456 and National Science Foundation of China Nos. 60703030, 60803146.

W. Qiu (✉) · B. Liu
School of Information Security Engineering, Shanghai Jiaotong University,
Shanghai, People's Republic of China
e-mail: qiuwd@sjtu.edu.cn

Z. Gong
Distributed and Embedded Security Group, Faculty of EEMCS,
University of Twente, Enschede, The Netherlands

Y. Long · K. Chen
Department of Computer Science and Engineering, Shanghai Jiaotong University,
Shanghai, People's Republic of China

Published online: 08 December 2009

1 Introduction

As the rapid development of the Internet, various technologies have been implemented to support online information systems, such as e-commercial or e-government systems [3,8,28,31]. Consequently, information security technologies especially cryptographic primitives play more and more important role in these online systems. A new logic ENDL (extension of non-monotonic dynamic logic) for verifying secure transaction protocols has been proposed in [8]. In the ENDL, traditional cryptographic tools, timestamps and RSA-based signed certificates, have been implemented for verifying the authentication properties of secure protocols and protecting data integrity. However, traditional finite field-based cryptosystems, such as Diffie-Hellman, RSA, and ElGamal, require the underlying field to be large enough to strengthen their security. While the knowledge information systems (such as e-commercial and e-government systems) are moving from the desktop to smart platforms such as cell phones or sensor nets, there is an increasing need of low-cost implementations. In resource-constrained applications, cryptosystems based on large finite fields are still impractical due to the limitations on the computation, storage and energy. Moreover, the known result [4] shows that a single bit communication consumes several orders of magnitude more power than computing a basic 32-bit arithmetic instruction. To reduce communication and storage costs, many typical cryptosystems [12,14,16,18,26] are proposed on the representation of the finite field elements with the counterparts of minimal polynomials. Due to Newton Identity [14], these cryptosystems can be looked the same as the systems based on characteristic sequences generated by linear feedback shift register (LFSR). For instance, LUC [26] can be represented as second-order sequence, while XTR [16] and GH [14] are generated by third-order sequence and Niederreiter [18] provides an encryption scheme and a key agreement protocol which are based on general nth order LFSR sequence. Based on these former work, Giulian and Gong [12] propose an ElGamal-like signature scheme on nth order characteristic sequences.

Electronic-cash, which was first introduced by Chaum [7], is designed for realizing untraceable payments in digital world. In an electronic cash (e-cash) system, anonymity and double-spending are the two major concerns. Privacy is always an important issue in electronic payment systems. Consumer's identity and personal information should be protected. It must be infeasible to discover a customer's identity or to trace an individual's purchasing activities. For electronic payment systems, anonymity is a necessity to protect customer's privacy. Some cryptographic tools, such as group signature and blind signature, are utilized to keep payment systems anonymous. Since digital information is easy to copy, the same e-cash spent more than once will be labeled as the double-spending problem. A practical e-cash system should thwart any double-spending customer, no matter whether he/she is malicious or not.

Partially blind signature, which was first introduced by Abe and Fujisaki [1], is used to solve the double-spending problem in e-cash applications. The scheme allows each of untraceable blind signatures to contain an explicit information on which both the signer and the user have agreed beforehand. For example, the signer can attach an expiry date and denomination to his/her blind signatures as an attribute. Accordingly, the attribute of the signature can be verified independently through the certified public key. This additional traceable property is very useful in privacy oriented e-services such as e-cash and e-voting systems. By following Abe and Fujisaki's initial work [1], many efficient partially blind signature schemes are presented. The schemes proposed in [23,30] are based on bilinear maps, but the verifying costs of pairing operation are not suitable for the applications where the client sides are limited in power durability, such as PDAs and sensors. In [29], Wu et al. present

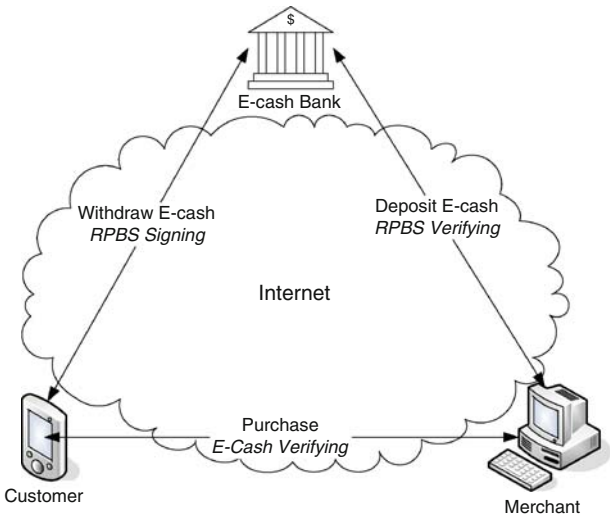


Fig. 1 A typical e-cash system architecture

two similar efficient schemes which are based on discrete logarithm problem (DLP). In [19], Okamoto proposes a less efficient but provably secure scheme in the standard model (which requires a slightly stronger computational complexity assumption, the 2SDH assumption, than the SDH assumption).

The partially blind signatures in literature [1,19,23,29,30] lack the restrictive property. The definition of restrictive blind signatures were proposed by Brands [6], which allows user to obtain a blind signature on a common message from signer, but the chosen message must conform to certain rules. Figure 1 shows how a restrictive partial blind signature (RPBS) can be applied in a e-cash systems. Partially blind signature can be used to sign and verify anonymous e-cash without double spending, whilst the restrictive property prevents the blind part of the message to be unacceptable or informal. There is no need to have different signing keys for different denominations due to the partial blind property. Based on blind Schnorr signature [21], Maitland and Boyd [17] first introduce a restrictive partially blind signature scheme with provable security. In [9,10], Chen et al. propose two restrictive partially blind signatures from bilinear maps.

Based on the above observations, this paper studies restrictive partially blind signature for resource-constrained information systems, and our primary goal is twofold. Firstly, we design two new LFSR sequence operations which are derived from the basic ones. And then we elaborate a new restrictive partially blind signature scheme, which is named S-RPBS, from nth order characteristic sequences generated by LFSR. Based on the computational hard assumptions given by Giulian and Gong [12], we show S-RPBS is provable secure in the random oracle model [5]. Since the efficient sequence operations and the reduced representation of finite field elements by LFSR, our scheme's computational costs are all directly reduced. Second, we propose an practical e-cash application based on S-RPBS. With a low-cost hardware implementation for LFSR sequence operations, the S-RPBS scheme will extremely benefit the privacy-oriented applications on highly constrained devices, such as cellular networks, tactical networks, and sensor nets.

The remainder of the paper is structured as follows. After some preliminary work in Sect. 2, we describe our S-RPBS scheme in Sect. 3, and then prove its security in Sect. 4,

and compare the performance with other related schemes in Sect. 5. Section 6 discusses a practical e-cash application based on S-RPBS. Section 7 concludes the paper.

2 Preliminaries

2.1 Partially blind signature

In the phase of partially blind signatures, signer and user are assumed to have agreed on a piece of common information, denoted by **info**, which might be sent from the user to the signer. Abe and Okamoto [2] formalized this definition by providing a function Ag. Function Ag is defined as a polynomial-time deterministic algorithm that completes the negotiation of **info** between the signer and the user correctly. Normally, the negotiation is considered to be done outside of the scheme.

Definition 1 (*Partially Blind Signature Scheme*). A partially blind signature scheme is a four-tuple $(\mathcal{G}, \mathcal{S}, \mathcal{U}, \mathcal{V})$.

- $\mathcal{G}$ is a probabilistic polynomial-time algorithm. $\mathcal{G}$ takes security parameter k and outputs a public and secret key pair (pk, sk).
- $\mathcal{S}$ and $\mathcal{U}$ are pair of probabilistic interactive Turing machines each of which has a public input tape, a private input tape, a private random tape, a private word tape, a private output tape, a public output tape, and input and output communication tapes. The random tape and the input tapes are read-only, and the output tapes are write-only. The private work tape is read-write. The public input tape of $\mathcal{U}$ contains pk generated by $\mathcal{G}(1^k)$, the description of Ag, and $\mathbf{info}_u$. The public input tape of $\mathcal{S}$ contains the description of Ag and $\mathbf{info}_s$. The private input type of $\mathcal{S}$ contains sk, and that for $\mathcal{U}$ contains a message **msg**. The lengths of $\mathbf{info}_s$, $\mathbf{info}_u$, and **msg** are polynomial in k. $\mathcal{S}$ and $\mathcal{U}$ engage in the signature issuing protocol and stop in polynomial-time. When they stop, the public output tape of $\mathcal{S}$ contains either completed or not-completed. If it is completed, then its private output tape contains common information **info**. Similarly, the private output tape of $\mathcal{U}$ contains either $\perp$ or (**info**, **msg**, **sig**).
- $\mathcal{V}$ is a polynomial-time algorithm. $\mathcal{V}$ takes (pk, **info**, **msg**, **sig**) and outputs either accept or reject.

Definition 2 (*Partial Blindness*). Let $\mathcal{U}_0$ and $\mathcal{U}_1$ be two honest users that follow the signature-issuing protocol. Let $\mathcal{S}^*$ play the following Game A in the presence of an independent umpire.

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$.
2. $(\mathbf{msg}_0, \mathbf{msg}_1, \mathbf{info}_{u_0}, \mathbf{info}_{u_1}, Ag) \leftarrow \mathcal{S}^*(1^k, pk, sk)$.
3. The umpire sets up the input tapes of $\mathcal{U}_0, \mathcal{U}_1$ as follows:
   - Selects $b \in_R \{0, 1\}$ and places $\mathbf{msg}_b$ and $\mathbf{msg}_{1-b}$ on the private input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$, respectively. b is not disclosed to $\mathcal{S}^*$.
   - Places $\mathbf{info}_{u_0}$ and $\mathbf{info}_{u_1}$ on the public input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$ respectively. Also place pk and Ag on their public input tapes.
   - Randomly selects the contents of the private random tapes.
4. $\mathcal{S}^*$ engages in the signature issuing protocol with $\mathcal{U}_0$ and $\mathcal{U}_1$ in a parallel and arbitrarily interleaved fashion. If either signature issuing protocol fails to complete, the game is aborted.

5. Let $\mathcal{U}_0$ and $\mathcal{U}_1$ output $(\mathbf{msg}_b, \mathbf{info}_{u_0}, \mathbf{sig}_b)$ and $(\mathbf{msg}_{1-b}, \mathbf{info}_{u_1}, \mathbf{sig}_{1-b})$, respectively, on their private tapes. If $\mathbf{info}_{u_0} \neq \mathbf{info}_{u_1}$ holds, then the umpire provides $\mathcal{S}^*$ with the no additional information. That is, the umpire gives $\perp$ to $\mathcal{S}^*$. If $\mathbf{info}_{u_0} = \mathbf{info}_{u_1}$ holds, then the umpire provides $\mathcal{S}^*$ with the additional inputs $\mathbf{sig}_b, \mathbf{sig}_{1-b}$ ordered according to the corresponding messages $\mathbf{msg}_0, \mathbf{msg}_1$.
6. $\mathcal{S}^*$ outputs $b' \in_R \{0, 1\}$. The signer S wins the game if $b' = b$.

A signature scheme is partially blind if, for every constant $c > 0$, there exists a bound k_0 such that for all probabilistic polynomial-time algorithm $\mathcal{S}^*$, $\mathcal{S}^*$ outputs $b' = b$ with probability at most $1/2 + 1/k^c$ for $k > k_0$. The probability is taken over the coin flips of $\mathcal{G}$, $\mathcal{U}_0, \mathcal{U}_1$, and $\mathcal{S}^*$.

Definition 3 (*Unforgeability*). Let $\mathcal{S}$ be an honest signer that follows the signature-issuing protocol. Let $\mathcal{U}^*$ play the following Game B in the presence of an independent umpire:

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$.
2. $Ag \leftarrow \mathcal{U}^*(pk)$.
3. The umpire places sk, Ag and a randomly taken $\mathbf{info}_s$ on the proper input tapes of $\mathcal{S}$.
4. $\mathcal{U}^*$ engages in the signature issuing protocol with $\mathcal{S}$ in a concurrent and interleaving way. For each **info**, let q_S be the number of executions of the signature issuing protocol where $\mathcal{S}$ outputs completed and **info** is on its output tapes. (For **info** that has never appeared on the private output tape of $\mathcal{S}$, define $q_S = 0$.)
5. $\mathcal{U}^*$ outputs a single piece of common information, **info**, and $q_S + 1$ signatures $(\mathbf{msg}_1, \mathbf{sig}_1), \ldots, (\mathbf{msg}_{q_S+1}, \mathbf{sig}_{q_S+1})$.

A partially blind signature scheme is unforgeable if, for any probabilistic polynomial-time algorithm $\mathcal{U}^*$ that plays the above game, the probability that the output of $\mathcal{U}^*$ satisfies

$$\mathcal{V}(pk, \mathbf{info}, \mathbf{msg}_j, \mathbf{sig}_j) = accept$$

for all $j = 1, \ldots, q_S + 1$ is at most $1/k^c$ where $k > k_0$ for some bound k_0 and some constant $c > 0$. The probability is taken over the coin flips of $\mathcal{G}, \mathcal{U}^*$, and $\mathcal{S}$.

In [6], Brands first defines the restrictive property for blind signature, which is described as follows. A similar property can be extended to partially blind signature [17].

Definition 4 (*Restrictiveness*). Let **msg** be such a message that the receiver knows a representation $(a_1, a_2, \ldots, a_k)$ of **msg** related to a generator set $(g_1, g_2, \ldots, g_k)$ at the start of a blind signature protocol. Let $(b_1, b_2, \ldots, b_k)$ represent the blind message $\mathbf{msg}'$ that the receiver obtains after the protocol has finished. If there exist two functions I_1 and I_2 such that

$$I_1(a_1, a_2, \ldots, a_k) = I_2(b_1, b_2, \ldots, b_k),$$

regardless of any message **msg** and the blind transformations applied by the receiver, then the protocol is called a restrictive blind signature protocol. The functions I_1 and I_2 are called blinding-invariant functions of the protocol with respect to $(g_1, g_2, \ldots, g_k)$.

The above are the necessary notions and definitions for restrictive partially blind signature. In the following section, we will give an introduction on the mathematical background of the LFSR sequences, which will be used in our scheme.

2.2 LFSR sequences and its reducing representations

Let q be a prime power, an irreducible polynomial $f(x)$ over $GF(q)$ can be represented by

$$f(x) = x^n - a_1 x^{n-1} + a_2 x^{n-2} - \cdots + (-1)^n a_n,$$

where all $a_1, \ldots, a_n \in GF(q)$. Let γ be a root of $f(x)$ in the extension $GF(q^n)$, and P is the order of γ. A sequence s, which is generated by a n-order LFSR over $GF(q)$, can be represented by the following recurrence:

$$s_{k+n} = a_1 s_{k+n-1} - a_2 s_{k+n-2} + \cdots + (-1)^{n+1} a_n s_k.$$

Let $\overline{s}_i = (s_i, s_{i+1}, \ldots, s_{i+n-1})$. For $i = 0, \ldots, P-1$, our initial state of $\overline{s}_i$ is given by $s_i = Tr(\gamma^i)$, where $Tr(.)$ is the trace map from $GF(q^n)$ to $GF(q)$. Thus the period of the sequence s_i is equal to the order of the root γ.

LFSR sequences are closely related to the minimal polynomials of finite field elements by Newton Identity [14]. We denote the set $A_k = (s_k, s_{2k}, \ldots, s_{tk})$, and observe that from the construction given by [12], we just need shorter representation length than the finite field case(for $n \cdot \log q$ bits to q^n); we can define length of A_k by

$$|A_k| = t \cdot \log q = \begin{cases} \frac{n}{2} \cdot \log q & \text{if } q = p^2, \text{ and } n \text{ is even} \\ \frac{n-1}{2} \cdot \log q & \text{if } q = p^2 \text{ and } n \text{ is odd} \end{cases}$$

For more details about the LFSR sequence and its reducing representation method, see the references [12,13,18,27].

2.3 Sequence operations

For a cryptosystem-based on LFSR sequences, Giulian and Gong [12] define two basic sequence operations.

Sequence Operation 1 (SO1): Given A_k and a random integer l, where $0 \leq k, l < P$, compute A_{kl}.

Sequence Operation 2 (SO2): Given state $\overline{s}_k$ and $\overline{s}_l$, where $0 \leq k, l < P$, compute $\overline{s}_{k+l}$.

SO1 can be calculated efficiently by the Fiduccia algorithm [11], while **SO2** can quickly be performed from the theory of LFSR sequences [13]. For the construction of our scheme in the next section, we present two new sequence operations **DSO1** and **DSO2** [15], which are derived from **SO1** and **SO2**.

Derived Sequence Operation 1 (DSO1): Given A_1 and an integer k, where $0 \leq k < P$, compute $\overline{s}_k$.

An efficient algorithm to execute **DSO1** can be constructed as follows:

1. Compute $A_k, A_{k+1}, \ldots, A_{k+n-1}$ from A_1 and k by **SO1**;
2. Obtain $s_k, s_{k+1}, \ldots, s_{k+n-1}$ from $A_k, A_{k+1}, \ldots, A_{k+n-1}$;
3. Output $\overline{s}_k = \{s_k, s_{k+1}, \ldots, s_{k+n-1}\}$.

Derived Sequence Operation 2 (DSO2): Given $\overline{s}_k$ and an integer l, where $0 \leq k, l < P$, compute $\overline{s}_{kl}$.

A feasible algorithm to compute **DSO2** is depicted as follows:

1. Compute $\overline{s}_k, \overline{s}_{2k}, \ldots, \overline{s}_{(t-1)k}$ from $\overline{s}_k$ by **SO2**;

2. Obtain $A_k = \{s_k, s_{2k}, \ldots, s_{(t-1)k}\}$ from $\overline{s}_k, \overline{s}_{2k}, \ldots, \overline{s}_{(t-1)k}$;
3. Output $\overline{s}_{kl} = \{s_{kl}, s_{kl+1}, \ldots, s_{kl+n-1}\}$ from A_k and l by **DSO1**.

Apparently, the two new sequence operations cost times of basic sequence operations (**SO1** and **SO2**), but we notice that all the sequence operations can be calculated by simple matrix operations [11,13]. We stress that the LFSR sequence operations can be optimized extremely from a low-cost hardware implementation [25]. For brevity, the feasible methods of LFSR sequence operations are detailed in [11–13,25].

2.4 Computational complexity problems

For the provable security of our sequence-based restrictive partially blind signature (S-RPBS), here the computational complexity assumptions under LFSR [12] are recalled.

Definition 5 (*Discrete Logarithm Problem* (*DLP*)): p is a large prime, g is a generator of group $\mathbb{Z}_p$, given $y = g^x (mod\ p)$, to compute $x = log_g y$.

Definition 6 (*State-Based Discrete Logarithm Problem* (*S-DLP*)): Given state $\overline{s}_1$ and $\overline{s}_l$, to determine l.

We notice that the recent algebra attack [22] can only be implemented if one uses LFSR as a pseudorandom function. In this work, we just use the property of a short representation and a feasible computation of LFSR sequences from $GF(q)$ to $GF(q^n)$. It was proven that solves the S-DLP is computationally equivalent to the DLP [12]. The security of S-RPBS scheme is based on assuming the intractability of the DLP.

3 Sequence-based restrictive partially blind signature scheme

The scheme of a sequence-based restrictive partially blind signature system consists of four phases: **Requesting**, **Signing**, **Extraction** and **Verifying**. A detailed description is as follows:

- **Entities**: signer $\mathcal{S}$ and receiver $\mathcal{U}$.
- **Domain parameters**: $\{q, n, P, A_1\}$. Let q be a prime power. A sequence s is generated by a n-order LFSR over $GF(q)$. P is prime order of the sequence. A_1 is the given sequence element set. M is an arbitrary message space, $\mathbf{m}$, $\mathbf{info} \in M$. $\mathcal{H}, \mathcal{F} : \{0,1\}^* \mapsto \mathbb{Z}_q$ denote two cryptographic hash functions.
- **Signer's Private key**: $x, 0 \leq x < P$.
- **Signer's Public key**: $\overline{s}_x$.

After the above initial work, our scheme processes with the following steps:

1. **Requesting**. Assume that $\mathcal{U}$ wants to get a partially blind signature on an implicit message $\mathbf{m} \in M$, and then $\mathcal{U}$ prepares an explicit common information $\mathbf{info} \in M$ and $z_1 = \mathcal{F}(\mathbf{m})$ that will be sent to $\mathcal{S}$ for his agreement. This negotiation can be done outside of the scheme. First, $\mathcal{S}$ computes $a_1 = z_1^r$ and $a_2 = z_1^x$. Next, $\mathcal{S}$ chooses a secure random number r, computes $\overline{s}_{rz_2}$ from A_1 and rz_2 by **DSO1**, where $z_2 = \mathcal{F}(\mathbf{info})$. Then $\mathcal{S}$ sends a_1, a_2 and $\overline{s}_{rz_2}$ to $\mathcal{U}$.

   After receiving $\overline{s}_{rz_2}$, $\mathcal{U}$ chooses two random numbers v, w and processes in the following steps:

   (a) Computes $\overline{s}_v$ from A_1, v by **DSO1**.

(b) Computes $\overline{s}_{wx}$ from $\overline{s}_x$, w by **DSO2**.
(c) Obtains $\overline{s}_{wx}, \overline{s}_{rz_2}, \overline{s}_v$, computes $\alpha = \overline{s}_{rz_2+v+wx}$ by **SO2**.
(d) After the sequences operations, $\mathcal{U}$ computes $C' = \mathcal{H}(\mathbf{m}||\mathbf{info}||\alpha)$, then sends $C = w - C' \ (mod\ P)$ and $b_1 = z_1^v, b_2 = a_2^w$ to $\mathcal{S}$.

2. **Signing**. After receiving C and **info**, $\mathcal{S}$ signs C with the randomizing factor r and his private key x, computes $D = r + z_2^{-1}Cx \ (mod\ P)$. Then $\mathcal{S}$ sends D to $\mathcal{U}$.

3. **Extraction**. After receiving $D, \mathcal{U}$ computes $D' = D + z_2^{-1}v \ (mod\ P)$. Hence, the resulting signature on m and **info** is a tuple $(a_1, a_2, b_1, b_2, C', D')$. The restrictiveness can publicly be confirmed from the following equation:

$$b_2 = \mathcal{F}(\mathbf{m})^{z_2 D'} \cdot a_2^{C'} \cdot a_1^{-z_2} \cdot b_1^{-1}. \tag{1}$$

4. **Verifying**. Given the signature $(\mathbf{m}, \mathbf{info}, C', D')$ on the public key $\overline{s}_x$, one can perform the following operations to check its correctness:

(a) Computes $\overline{s}_{z_2 D'}$ from $A_1, z_2 D'$ by **DSO1**.
(b) Computes $\overline{s}_{C'x}$ from $\overline{s}_x, C'$ by **DSO2**.
(c) Computes $\overline{s}_{z_2 D' + C'x}$ from $\overline{s}_{z_2 D'}, \overline{s}_{C'x}$ by **SO2**.
Because $D' = z_2^{-1}v + D$, $C' = w - C$ and $D = r + z_2^{-1}C \cdot x$, we can easily get

$$\begin{aligned} z_2 D' + C'x &= v + z_2 D + C'x \\ &= v + rz_2 + (C + C')x \\ &= rz_2 + v + wx. \end{aligned} \tag{2}$$

In addition to the restrictiveness, one accepts the signature as valid if it satisfies the following equation:

$$\mathcal{H}(\mathbf{m}||\mathbf{info}||\overline{s}_{z_2 D' + C'x}) = \mathcal{H}(\mathbf{m}||\mathbf{info}||\alpha) = C'.$$

This ends the description of S-RPBS signature scheme. Figure 2 gives an illustration of S-RPBS. Besides some hash and simple modular operations, the full scheme (signing and verifying) only requires seven times sequence operations. We note that the LFSR sequence operations require low computation and storage costs on either hardware or software implementation [25].

4 Security analysis

In this section we will formally analyze the security of S-RPBS scheme. For the sake of simplicity, the definitions and notions shown below are the same as Sect. 3. With $\mathcal{H}, \mathcal{F} : \{0, 1\}^* \mapsto \mathbb{Z}_q$ denoting two cryptographic hash functions, we have $z_1 = \mathcal{F}(\mathbf{m})$ and $z_2 = \mathcal{F}(\mathbf{info})$ for messages **m** and **info**, respectively.

4.1 Partial blindness

Here we follow Game A, which is mentioned in Sect. 2.1, to prove that S-RPBS scheme satisfies the partially blind property. For each instance i of the proposed scheme, signer $\mathcal{S}^*$ can record C_i received from $\mathcal{U}$ who communicates with $\mathcal{S}^*$ during the ith instance of the scheme. The tuple (C_i, D_i) is usually referred to the view of $\mathcal{S}^*$ to the instance i. Thus, we have the following theorem:

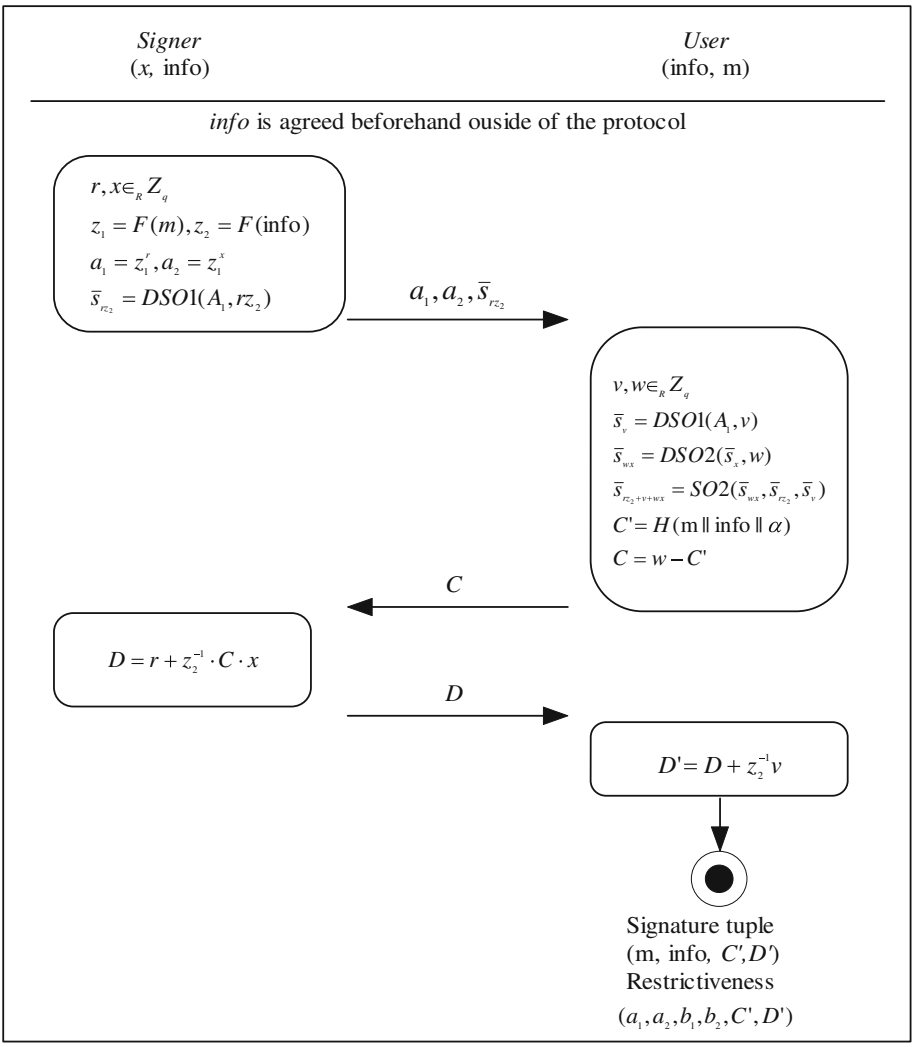


Fig. 2 An illustration of the S-RPBS scheme

Theorem 1 *S-RPBS scheme is partially blind.*

Proof Since the tuple (**m**, **info**, C', D') is produced by the proposed scheme, we have $C' = w - C_i$, $D' = D_i + z_2^{-1} v$ and $D_i = rz_2 + C_i x$ under the same (**m**, **info**). From the view of $\mathcal{S}^*$, since v, w are two random numbers selected by $\mathcal{U}$, and $|v| = |D_i|$, $|w| = |C_i|$. The existence of two random values (v, w) protects (C', D') under information-theoretic security. Hence $\mathcal{S}^*$ cannot derive (v, w) from each view of (C_i, D_i), $C' = \mathcal{H}(\mathbf{m}||\mathbf{info}||\bar{s}_{z_2 D' + C' x})$ is satisfied while the instance $i \mapsto \{0, 1\}$, therefore, following Game A, even an unbounded powerful $\mathcal{S}^*$ can succeed in determining i with probability 1/2, the theorem, follows. □

From the proof of Theorem 1, we can realize the importance of random factors v, w. $\mathcal{U}$ must update v, w in each of the new instance. The used random factors v, w must be destroyed after the signature (**m**, **info**, C', D') is created.

4.2 Restrictiveness

Based on Definition 4 in Sect. 2.1, we briefly prove the restrictive property of S-RPBS scheme.

Theorem 2 *S-RPBS scheme is restrictive.*

Proof Assume that the tuple ($\mathbf{m}$, **info**, C', D') is produced by the S-RPBS scheme; then we have $D' = D_i + z_2^{-1}v, C' = w - C_i$ and $D_i = rz_2 + C_i x$ under the message $\mathbf{m}$ and the common information **info**. From the Eq. (2) in Sect. 3, the restrictiveness holds if we find that

$$\begin{aligned} b_2 &= z_1^{z_2 D'} \cdot a_2^{C'} \cdot a_1^{-z} \cdot b_1^{-1} \\ z_1^{wx} &= z_1^{z_2 D' + C' x} \cdot z_1^{-rz_2} \cdot z_1^{-v} \end{aligned} \tag{3}$$

Due to the Eq. (1) in Sect. 3, we have

$$\begin{aligned} \mathcal{F}(\mathbf{m})^{wx} &= z_1^{z_2 D' + C' x} \cdot z_1^{-rz_2} \cdot z_1^{-v} \\ &= \mathcal{F}(\mathbf{m})^{rz_2 + v + wx} \cdot \mathcal{F}(\mathbf{m})^{-rz_2} \cdot \mathcal{F}(\mathbf{m})^{-v} \\ &= b_2 \end{aligned} \tag{4}$$

Since the DLP is intractable, no one can either obtain the random factors r, v and w from a_1, b_1, or recover the secret key x from a_2, b_2. Based on the above analysis, It is easy to see our scheme also achieves the restrictive property which is attributed to Maitland and Boyd's restrictive partially blind signature [17]. So, the theorem holds. □

4.3 Unforgeability

Now we discuss the unforgeability of S-RPBS in the random oracle model by assuming the intractability of the DLP.

Theorem 3 *If there exists a forger $\mathcal{U}^*$ who can forge a valid signature of S-RPBS scheme in polynomial time-bound t with non-negligible probability ϵ, then there exists an algorithm $\mathcal{A}$ that solves the S-DLP problem in polynomial time-bound t' with non-negligible probability ϵ'.*

Proof Suppose a forger $\mathcal{U}^*$ can forge a valid signature ($\mathbf{m}$, **info**, C', D') in (t, ϵ). By exploiting $\mathcal{U}^*$, we can construct an algorithm $\mathcal{A}$ that solves S-DLP problem in (t', ϵ'). $\mathcal{A}$ has two random oracles $\mathcal{H}, \mathcal{F} : \{0,1\}^* \mapsto \mathbb{Z}_q$ to answer hashing queries.

Let q_H, q_F be the maximum number of queries that $\mathcal{U}^*$ can ask from $\mathcal{H}$, $\mathcal{F}$, and q_S be the maximum number of signing queries. From $i \in \{1, 2, \ldots, q_H\}$, $\mathcal{M}$ chooses $r_i \in_R \mathbb{Z}_q^*$ and sends $\bar{s}_{r_i}$ to $\mathcal{U}^*$. $\mathcal{U}^*$ computes α_i, then sends the tuple ($\mathbf{m}_i$, $\mathbf{info}_i$, α_i) to the random oracle $\mathcal{H}$ for computing its hash value $\mathcal{H}(\mathbf{m}_i || \mathbf{info}_i || \alpha_i)$. Let $z_{1,i} = \mathcal{F}(m)$ and $z_{2,i} = \mathcal{F}(\mathbf{info}_i)$. Let q_F be the maximum number of queries that $\mathcal{U}^*$ can ask from the random oracle $\mathcal{F}$. It is easy to prove that the simulation is successful due to the indistinguishability [2].

Then $\mathcal{A}$ use $\mathcal{U}^*$ to solve S-DLP. After the above training, $\mathcal{U}^*$ can get a valid signature tuple (α_1, D_1', C_1') in (t, ϵ). Because $\mathcal{U}^*$ can only get hash values from $\mathcal{H}$. Next, by using the forking lemma reduction [20], $\mathcal{M}$ repeats with the same random tape and a different choice of $\mathcal{H}$. $\mathcal{U}^*$ can get another valid signature (α_2, D_2', C_2') after polynomial time t with the same

probability at least ϵ. Since $\alpha_1 = \alpha_2$ with a non-negligible probability $1/\sqrt{q_H}$, from the equation $C' = \mathcal{H}(\mathbf{m}||\mathbf{info}||\overline{s}_{zD'+C'x})$, we have

$$\overline{s}_{z_{2,1}D'_1+C'_1x} = \overline{s}_{z_{2,2}D'_2+C'_2x},$$

then we have the equation

$$z_{2,1}D'_1 + C'_1 \cdot x = z_{2,2}D'_2 + C'_2 \cdot x.$$

Because $\mathcal{H}$ was uniformly changed choices in the second run, both $D'_1 \neq D'_2$ and $C'_1 \neq C'_2$ have a overwhelming probability; $\mathcal{M}$ can compute x from

$$x = \frac{(z_{2,2}D'_2 - z_{2,1}D'_1)}{(C'_1 - C'_2)}.$$

Because $\mathcal{A}$ only knows the public key $\overline{s}_x$, from the above equation, $\mathcal{A}$ can derive x from $\overline{s}_{z_2D'+C'x}$. It means $\mathcal{A}$ solves S-DLP problem in polynomial time $t' \approx 2t$ with non-negligible probability $\epsilon' \approx \frac{\epsilon^2}{q_S^2 \cdot \sqrt{q_H}}$. □

Because it was proven that to solve S-DLP is equivalent to DLP [12], the forger faces the problem as hard as solving DLP problem. From Theorem 2, we have the following theorem:

Theorem 4 *If there exists an adaptive attacker that (t, ϵ)-breaks S-RPBS scheme, it can be converted to an algorithm $\mathcal{A}$ that solves DLP problem in polynomial time $t' \approx 2t$ with success probability $\epsilon' \approx \frac{\epsilon^2}{q_S^2 \cdot \sqrt{q_H}}$.*

5 Performances

Here, we analyze the performance of S-RPBS with some related restrictive partially blind signature schemes, such as [9,10,17]. According to the existing result [25], one sequence operation takes nearly the same time as one modular exponentiation in software realization, but will be faster in the hardware registers. Also from the related experiments, one pairing operation is at least 3–4 times more expensive than a modular exponentiation. The verifying costs of pairing are not suitable for the applications where the client sides have constrained resource, such as PDAs and sensors. It was analyzed by Sin [25] that the 1024-bit security level of DLP over $GF(q^n)$ needs 340 bits representation of LFSR over $GF(q)$. Since S-RPBS is based on the intractability of the DLP over $GF(q^n)$, it also keeps the efficiency of the shorter representation of the security parameters. According to our experiment,[1] the performance of the shorter security parameters in LFSR sequence operations is given in Table 1.

In Table 2, we compare S-RPBS to other related restrictive partially blind signature schemes [9,10,17]. In theory, our proposed scheme shows its advantages of the computational costs among the related partially blind signature schemes with the restrictive property.

Based on our implementation, Table 3 shows a detailed performance of S-RPBS. Consistent with the theoretical analysis, the result in Table 3 supports that S-RPBS is time- and storage-efficient for realizing e-cash systems in practice. We stress that the LFSR sequence operations can be optimized extremely from a low-resource hardware implementation [25].

[1] A Windows Vista Business PC with Intel Core2 Duo T7250 CPU and 2,048 MB of RAM is used for the implementation.

Table 1 LFSR sequence operations process time on PC in C using NTL [24]

LFSR sequence operation (340 bits)	Max time (min)	Min time (min)	Average time (min)
SO1	306.6	77.8	94.7
SO2	3.8	1.7	2.0
DSO1	638.7	405.3	488.8
DSO2	1,055.3	172.9	218.3

Table 2 Theoretical performances of some related restrictive partially blind signature schemes

	S-RPBS	Maitland02 [17]	Chen06 [9]	Chen07 [10]
Mathematical foundation	LFSR	DLP	Pairing	Pairing
Necessary random numbers	3	9	4	5
Signing costs	$4T_{so}$	$15T_e$	$3T_p + 10T_e$	$5T_p + 9T_e$
Verifying costs	$3T_{so}$	$6T_e$	$5T_p$	$3T_p + 2T_e$
Signature sizes	4ℓ	4ℓ	4ℓ	4ℓ

T_{so} time for one sequence operation, T_e time for one exponentiation computation, T_p time for one bilinear pairing, Typical secure length: $\ell = 160$ bit

Table 3 A detailed performance of S-RPBS

S-RPBS	Theoretical costs	Average experiment result (min)
Signing costs	$4T_{so}$	1,119.6
Verifying costs	$3T_{so}$	723.5
Total costs	$7T_{so}$	1,878.6

6 S-RPBS application for electronic cash

Electronic cash (E-cash) systems are similar to currencies used in the real-world. As shown in Fig. 1, three major parts are involved in a typical e-cash system, which includes a customer, a merchant, and an e-cash bank. To ensure the security of the e-cash system, the following main protocols are required to be implemented:

1. Initialization. The protocol that initializes and distributes the system parameters to the participant of an e-cash system.
2. Withdraw. The protocol that allows a customer or merchant with e-cash to withdraw "real" money from banks;
3. Payment. A customer browsers a merchant's site to purchase the items with the payment protocol;
4. Deposit. A merchant can carry out the deposit protocol with e-cash bank to be credited.

According to whether the bank is involved in the payment protocol or not, an e-cash system can be further divided into *on-line* and *off-line* models. The major difference between on-line and off-line systems is that the double-spending behavior can be detected immediately in an on-line system. Some additional cryptographic technologies have to be deployed to find the identity of double-spending cash's owner under the off-line model. However, if there exists

Table 4 List of symbols used in the e-cash application

Symbol	Notation
q	a prime order
n	a LFSR's order over $GF(q)$
P	Prime order of a LFSR sequence
A_1	a given sequence element set
msg	a description of merchant's ID and selected items
info	a description of prices and serial numbers
H, F	two cryptographic hash functions
C', D'	two elements of S-RPBS signature output:(**m**, **info**, C', D')

a huge number of purchases simultaneously, on-line double-spending check might become the bottleneck of the whole e-cash system.

In the rest of this section, an online PDA (or smartphone) based e-cash system will simply be depicted to show how the time and storage efficiency of S-RPBS become the certain advantages on the implementation of an e-cash system. In the system, **info** denotes the description of price and serial numbers which will be revealed in the signing process. This partial and restrictive properties ensure the anonymity of the e-cash system and prevent the system from double-spending. The details will be explained later in following subsection. For the ease of reading, a summarization of symbols used in the description are summarized from S-RPBS in Table 4.

6.1 The initialization

Because of customer and merchant needs to establish an account with the e-cash bank and implements a client software, the initialization of the e-cash system includes the following parameters which were introduced in Sect. 3 as well.

1. q, n, P, A_1. A_1 is the given sequence element set. P is prime order of the sequence.
2. Let M be the message space, where **msg**, **info** $\in M$. **msg** denotes the description of a merchant's ID and other selected items. **info** denotes the description of price and serial numbers which will be revealed in the signing process.
3. $H, F : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ denote two cryptographic hash functions which will be used in the scheme.
4. E-cash bank's private key: x where $x \leftarrow \mathbb{Z}_P$. The corresponding public key is $\overline{s}_x$.

6.2 The withdrawal protocol

The customer browses the merchant's website via the software client installed on his/her PDA or smartphone, chooses items he/she wants to purchase. Then the customer generates **msg** and **info**. The serial number is created in such a way that the possibility of every customer generating the same serial number is minimal. After that, the customer and the e-cash bank will carry out the request and signing process via the S-RPBS scheme (which is described in Sect. 3) to obtain the e-cash as the final signature {**msg**, **info**, C', D'}. The bank will search in the local database to check whether the received serial number was used earlier. If it is duplicated, the bank will notify the customer to choose another serial number to avoid a

double spending. Finally, the e-cash bank will debit the amount of the corresponding price from the customer's account and finish the withdrawal transaction.

6.3 The payment protocol

After getting a certain amount of e-cash from the bank, the customer pays the e-cash to the merchant. The merchant first checks the tuple {**msg**, **info**} contained in the coin to make sure the consistency of the purchase and verifies the signature by following the verification process in S-RPBS.

6.4 The deposit protocol

The merchant sends the tuple {**info**, C', D'} to the e-cash bank online. The message **msg**, which includes the purchasing information, is blinded to the bank to protect the privacy of the customer. Thus a customer's purchase activities can remain unknown to the e-cash bank. The e-cash bank searches the serial number in the local deposited database. If there exists a duplicated serial number in the database, the e-cash bank returns a "double spending" message to the merchant. So the merchant will reject the purchase request. Otherwise, the bank will send a valid message to the merchant to confirm the payment. After receiving the confirmation, the merchant will make sure the purchase is valid and delivers the items to the customer. Then the whole purchase process is done.

The above application is a simple e-cash system based on our S-RPBS scheme. The application enjoys the anonymity and thwarts the double-spending problem as well. Nevertheless, according to the time and storage efficiency of S-RPBS on the server side, the hardware requirement of information systems for bank servers will be lower than other partially blind signature schemes based on large number fields.

7 Conclusion

In this paper, we have proposed a new restrictive partially blind signature scheme which is based on nth order characteristic sequences generated by LFSR and proven its security in the random oracle model. Compared to the existing finite field-based schemes, our sequence-based scheme is resource-efficient because the special properties of the LFSR sequences. The efficiency will benefit the applications, such as e-cash and e-voting, in resource-constrained environments. Based on the our initial work, more sequence-based cryptosystems for low-resource implementations might be designed or analyzed on two derived LFSR sequence operations.

References

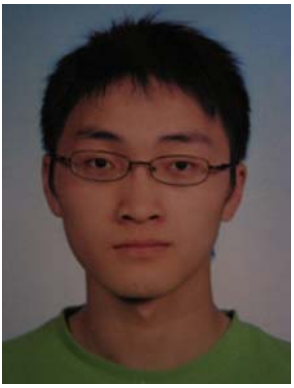
1. Abe M, Fujisaki E (1996) How to date blind signatures. In: Advances in Cryptology-ASIACRYPT'96, LNCS 1163, pp 244–251
2. Abe M, Okamoto T (2000) Provably secure partially blind signatures. Advance in Cryptology-CRYPTO'00, LNCS 1880, pp 271–286
3. Albers M, Jonker CM, Karami M, Treur J (2004) Agent models and different user ontologies for an electronic market place. J Knowl Inf Syst 6(1):1–41
4. Barr KC, Asanovic K (2006) Energy aware lossless data compression. ACM Trans Comput Syst 24(3):250–291

5. Bellare M, Rogaway P (1993) Random oracles are practical: a paradigm for designing efficient protocols. In: ACM CCS'93, pp 62–73
6. Brands S (1993) An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI, March 1993
7. Chaum D (1983) Blind signature for untraceable payments. In: Advances in Cryptology-CRYPTO'82, pp 199–203
8. Chen QF, Zhang CQ, Zhang SC (2005) A logical framework for verifying secure transaction protocols. J Knowl Inf Syst 7(1):84–109
9. Chen XF, Zhang F, Mu Y, Susilo W (2006) Efficient provably secure restrictive partially blind signatures from bilinear pairings. In: Financial Cryptography and Data Security 2006, LNCS 4107, pp 251–265
10. Chen XF, Zhang F, Liu S (2007) ID-based restrictive partially blind signatures and applications. J Syst Softw 80(2):164–171
11. Fiduccia CM (1985) An efficient formula for linear recurrences. SIAM J Comput 14:106–112
12. Giulian KJ, Gong G (2004) New LFSR-based cryptosystems and the Trace discrete log problem (Trace-DLP). SETA 2004, LNCS 3486, pp 298–312
13. Golomb S (1982) Shift register sequences. Aegean Park, Laguna Hills
14. Gong G, Harn L (1999) Public-key cryptosystems based on cubic finite field extensions. IEEE Trans IT 24:2601–2605
15. Gong Z, Long Y, Chen K (2007) Efficient partially blind signature from LFSR. In: SNPD 2007, IEEE Computer Society Proceedings, August 2007, pp 717–722
16. Lenstra A, Verheul E (2000) The XTR public key System. In: Advances in Cryptology-Crypto 2000, LNCS 1880, pp 1–19
17. Maitland G, Boyd C (2002) A provably secure restrictive partially blind signature scheme. In: PKC 2002, LNCS 2274, pp 99–114
18. Niederreiter H (1993) Finite fields and cryptology. Finite fields, coding theory, and advances in communications and computing. M.Dekker, New York, pp 359–373
19. Okamoto T (2006) Efficient blind and partially blind signatures without random oracles. In: Halevi S, Rabin T (eds) TCC 2006, LNCS 3876, pp 80–99
20. Pointcheval D, Stern J (2000) Security arguments for digital signatures and blind signatures. J Cryptology 13(3):361–396
21. Schnorr CP (1991) Efficient signature generation by smart cards. J Cryptology 4(3):161–174
22. Shamir A (2008) Cube attacks on tweakable black box polynomials. In: Crypto'08, invited talk
23. Chow SSM, Hui LCK, Yiu SM, Chow KP (2005) Two improved partially blind signature schemes from bilinear pairings. In: Boyd C, Gonzalez Nieto JM (eds) ACISP 2005, LNCS 3574, pp 316–328. Full version at Cryptology ePrint Archive, Report 2004/108
24. Shoup V NTL: a library for doing number theory. http://www.shoup.net/ntl/WinNTL-5_5_1.zip
25. Sin S GH-PKC software implementation. http://comsec.uwaterloo.ca/projects.html#gh
26. Smith P, Skinner C (1994) A public-key cryptosystem and a digital signature system based on the lucas function analogue to discrete logarithms. In: Advances in Cryptopogy-Asiacrypt'94, LNCS 917, pp 357–364
27. Tan C, Yi X, Siew C (2003) On the n-th order shift register based discrete logarithm. IEICE Trans Fundam E86-A:1213–1216
28. Tran T (2009) Protecting buying agents in e-marketplaces by direct experience trust modelling. J Knowl Inf Syst (OnlineFirst, Jan 2009)
29. Wu Q, Susilo W, Mu Y, Zhang F (2006) Efficient partially blind signatures with provable security. In: Gavrilova M et al (eds) ICCSA 2006, LNCS 3982, pp 345–354
30. Zhang F, Safavi-Naini R, Susilo W (2003) Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In: Cryptology-INDOCRYPT 2003, LNCS 2904, pp 191–204
31. Zhuang Y, Fong S, Shi M (2008) Knowledge-empowered automated negotiation system for e-Commerce. J Knowl Inf Syst 17(2):167–191

Author Biographies

Weidong Qiu got his MS degree in Cryptography from Xidian University in 1998 and PhD degree in Computer Software Theory from Shanghai Jiaotong University in 2001. Before he came to Shanghai Jiaotong University in 2004, he had been working as a postdoc in Hagen FernUni of Germany (2001–2003). He is now an Associate Professor and head assistant at the School of Information Security and Engineering. Dr. Qiu is the editor of the book "Cryptographic Protocols" (China Higher Education Press). His main research area includes cryptographic theory, technology of network security and computer forensic. He has published tens academic papers on cryptology.

Zheng Gong received his MS degree in Computer Science and Technology in South China University of Technology in 2002, and received the PhD degree in Computer Science and Technology in Shanghai Jiaotong University in 2008. Since December 2008, he has been a postdoc of DIES group in University of Twente, Netherlands. His recent research directions are cryptography and provable security in distributed and embedded systems, especially lightweight cryptographic primitives and protocols for resource-constrained environments.



Bozhong Liu got his BS degree in Information Security from Shanghai Jiaotong University in 2005. He is now a PhD Candidate of School of Information Security and Engineering in Shanghai Jiaotong University who majors in Computer Science and Technology (cryptography and computer security). He is a member of Cryptography and Information Security Lab and his current research interests include dictionary research, hash functions applications, applied cryptographic techniques such as privacy-preserved data mining and computer security.

Yu Long received his PhD degree in the Department of Computer Science and Engineering in Shanghai Jiao Tong University, 2008, and now she is a research assistant in the same school. Her research interests include information theory and modern cryptography, etc.

Kefei Chen received his BS and MS degrees from Northwest Telecommunications Engineering Institute, Xi'an, in 1982 and 1985, respectively. He was awarded the PhD degree from Justus-Liebig University, Germany, in 1994. He is currently a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai. He is the Director of Cryptography and Information Security Lab at Shanghai Jiao Tong University, and involved in Expert Panel Committee in the Department of Information Sciences of NSFC. His current research interests include public key cryptography, cryptographic protocol analysis and automatic verifying, applied cryptographic techniques and computer security.