









# Traceable Ring Signature Schemes Based on SM2 Digital Signature Algorithm and Its Applications in the Evidence-Storage System

Yongxin Zhang<sup>1</sup> , Qinghao Wang<sup>1,2</sup> , Ning Lu<sup>2,3</sup> , Wenbo Shi<sup>2</sup> ,  
and Hong Lei<sup>1,4</sup>  

<sup>1</sup> SSC Holding Company Ltd., Chengmai 571924, China

{yongxin, qinghao}@oxhainan.org, leiluono1@163.com

<sup>2</sup> The College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China

{luning, shiwb}@neuq.edu.cn

<sup>3</sup> The School of Computer Science and Technology, Xidian University, Xi'an 710071, China

<sup>4</sup> The School of Cyberspace Security, Hainan University, Haikou 570228, China

**Abstract.** A high-quality evidence-storage service is crucial for many existing applications. For example, judicial or arbitral authorities need to guarantee that their systems are available and trustworthy to conduct the arbitration. Such a system should protect witnesses' privacy from potential adversary threats. Ring signatures can be employed in blockchain-based systems to conceal the witness's identity among a group of persons while guaranteeing the availability and trustworthy of evidence. However, the strong anonymity of ring signature makes regulation tough and shields criminals. The traceable ring signature (TRS) is a de-anonymization mechanism that, unlike group signatures, does not rely on centralized trust, making it suitable for the blockchain system. Unfortunately, no SM2-based designs could be discovered in the TRS public literature. To fill the gap, this paper proposes a traceable ring signature scheme based on SM2 digital signature algorithm. It is shown that SM2 traceable ring signature (STRS) satisfies integrity, unforgeability, anonymity, and traceability. Moreover, we present an STRS-based blockchain evidence-storage system, in which users upload evidence with traceable ring signature generated by themselves, and regulators can learn the true identity of the signer if necessary.

**Keywords:** Blockchain · Evidence-storage system · SM2 · Traceable ring signature · Traceability

## 1 Introduction

Ring signatures, introduced by Kalai, Rivest, and Shamir in [1], allow a participant to sign a message anonymously on behalf of a group, named "ring". The

verifier can check the validity of the signature but cannot determine the identity of signer within the ring; this attribute is also possessed by group signature. However, the major difference between ring signatures and group signatures [2] is that there is no centralized manager who generates the keys, manages the group, and de-anonymizes the signer if required. This type of digital signature, which ensures anonymity without a centralized manager, is therefore appropriate for a decentralized system, and there are existing uses for it as a building block. For example, each transaction in blockchain clearly shows the input and output accounts (public keys) of funds, hence, we can determine the flow direction of funds. Monero requires user's key and other public keys to form a ring as the input account to construct the transaction, preventing the external observers from learning which possible signer in the ring is the real input account, thus realizing untraceability [3–5].

However, the lack of manager in ring signatures enables members to abuse their anonymity. In the example of Monero [6], the strong anonymity provides shelter for illegal acts, such as money laundering [7]. Due to the anonymity guarantees, the regulators cannot distinguish whether this transaction is signed by which one.

To address the uncontrolled anonymity guaranteed by ring signatures, Fujisaki and Suzuki proposed traceable ring signatures (TRS) in [8], in which each message is attached with a one-time random number, named ISSUE. TRS contains an algorithm Trace that can detect the signer in the ring. In detail, if Alice signed two messages  $m$  and  $m'$  to get traceable ring signatures  $\sigma$  and  $\sigma'$  with the same ISSUE, the Trace will expose the public key of Alice. It should be noted that if the message  $m = m'$ , it just finds the same signer but cannot tell who signed them, which is known as linkability [9].

Fujisaki [10] presented a sub-linear traceable ring signature with the trusted common reference string (CRS). Ho Au et al. in [11] build the traceable ring signatures based on bilinear maps. Scafuro et al. [12] proposed the one-time traceable ring signatures, where a member can sign anonymously only one message. Fan et al. [13] presented a ring signature with SM2 algorithm, where SM2 (Shangyong Mima) is proposed by the State Password Administration of China. Peng et al. [14] constructed the ring signature with SM9 algorithm. Although TRS has several variants, no scheme has yet been developed that is based on SM2, the Chinese cryptographic public key algorithm standard [15]. To balance the privacy and regulation of signers, we propose a traceable ring signature scheme based on the SM2 signature algorithm in this paper. In our proposed scheme, the signer can generate a valid ring signature that can be verified efficiently in a manner of privacy-preserving. Moreover, all signers can combine to trace back to the signer of a certain signature without revealing the identities of the rest.

## 1.1 Application Case

**Regulated Anonymous Evidence-Storage System.** Born with open, transparent, and tamper-proof qualities, blockchain technology is ideal for the

evidence-storage system to provide the solidification and permanent maintenance of data [16–18]. In the case of a disagreement, users, judicial or arbitration institutions can access proof from any node of the blockchain, eliminating the requirement for a third-party institution to give proof and thereby increasing the efficiency of related work. It is a challenge to settle out how evidence submitters can maintain their identity privacy for safe while yet proving their identification to the judiciary when necessary without alerting a third party.

Based on our scheme, the user first generates the ring signature for the evidence, after which others may check it and discover it is signed in a ring. Upon receiving the trace requirements, the user resigins a new message. Finally, the regulator can determine who the true signer of the signature is.

## 1.2 Our Contributions

To protect the user’s privacy, we propose a novel traceable ring signature scheme based on SM2 signature algorithm, which yields privacy and traceability. Our key contributions in this paper are as follow:

- We design a traceable ring signature scheme based on SM2 signature algorithm.
- We prove that our scheme can satisfy the property of privacy-preserving and traceability.
- We construct a blockchain evidence-storage scheme based on our algorithm and design a traceable data structure used in the tracing process, proving the feasibility of proposed scheme.

## 1.3 Layout

The rest of this paper is organized as follows. In Sect. 2, we give the preliminaries in this paper. In Sect. 3, we show the construction of the proposed SM2 traceable ring signature (STRS). In Sect. 4, we analyze the cost of STRS. In Sect. 5, we propose a blockchain evidence-storage system (SBES) based on STRS. In Sect. 6, we analyze the properties of SBES. We then present the conclusion in the last section.

# 2 Preliminaries

## 2.1 Notions

In this paper, we set  $\lambda$  represents the security parameter,  $\varepsilon$  denotes negligible function,  $\mathcal{PPT}$  represents the probability polynomial time,  $H$  represent the hash algorithm, e.g., SM3 cryptographic hash algorithm [19]. Also,  $\mathbb{G}$  is an elliptic curve point group of order  $q$ ,  $G$  is the generator of  $\mathbb{G}$ .

## 2.2 SM2 Signature Algorithm

SM2 is a public key encryption standard adopted by the People's Republic of China. SM2 Public key cryptography algorithms based on elliptic curve mainly include a trio of parts: digital signature algorithm, key exchange protocol and public key encryption.

In this section, we briefly review the SM2 digital signature algorithm, which includes a set of algorithms: Setup, Key Generation, Signature Generation and Verification, defined below:

1. **Setup.** Given the security parameter  $1^\lambda$ , the algorithm outputs an elliptic curve point group  $\mathbb{G}$  of order  $q$ , where  $G$  is the generator of  $\mathbb{G}$  and a hash function  $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ .
2. **Key Generation.** A user  $U$  randomly chooses  $d_A \in \mathbb{Z}_q^*$  as the private key, sets the public key  $P_A = d_A \cdot G$ . The algorithm outputs the key pair  $(pk, sk)$ .
3. **Signature Generation.** Given a user's private key  $d_A$ , a message  $m$ , the user  $A$  first computes  $e = H_1(m)$ , then randomly chooses  $k \in \mathbb{Z}_q^*$ , then calculates  $(x_1, y_1) = k \cdot G$ , computes  $r = (x_1 + e) \bmod q$  and  $s = ((1 + d_A)^{-1} \cdot (k - r \cdot d_A)) \bmod q$ . Finally, the algorithm outputs the signature  $\sigma = (r, s)$ .
4. **Verification.** Given a user's private key  $d_A$ , a message  $m$ , a signature  $(r, s)$ , the verifier first checks whether  $r, s \notin \mathbb{Z}_q^*$ , then computes the hash value  $e = H_1(m)$ , lets  $t = (r + s) \bmod q$ , calculates  $(x_1, y_1) = s \cdot G + t \cdot P_A$ , sets  $R = (e + x_1) \bmod q$ . If  $R = r$ , the algorithm outputs 1, which means the signature is valid; otherwise outputs 0.

## 3 The Proposed SM2 Traceable Ring Signature

In this section, we put forward the SM2 traceable ring signature (STRS). Generally speaking, a traceable ring signature works as follows: Each user  $U_i$  generates its own key pair  $(pk_i, sk_i)$ . They register their own public key  $P_i$  with the public key infrastructure. The user  $U_i$  randomly chooses  $(n - 1)$  users' public keys, adds his own public key, constructs the list  $\text{LIST} = \{pk_1, pk_2, \dots, pk_n\}$ . For a topic  $\text{ISSUE}$ , the user forms a tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ , he can sign a message  $m$  with his private key and the tag. If the signer outputs two message/signature pairs  $(i, m, \sigma)$ ,  $(i', m', \sigma')$ , if  $i \neq i'$ , everyone can know that the two signatures are independent; else if  $i = i'$ ,  $m = m'$ , we can link the two signatures; otherwise, we can trace the identity of the real signature.

### 3.1 Definition

A traceable ring signature scheme [8]  $\Sigma$  is a quartet of algorithms  $\{\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Trace}\}$  defined as follows.

- **KeyGen.** It is a probabilistic polynomial-time algorithm, which takes as input the security parameter  $1^\lambda$ , outputs a key pair  $(pk, sk)$ .

- **Sign.** It is a probabilistic polynomial-time algorithm, which takes as input a private key  $sk_i$  where  $i \in [1, n]$ , a tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$  where  $\text{LIST} = \{pk_1, pk_2, \dots, pk_n\}$ , and a message  $m$ , outputs a signature  $\sigma$ .
- **Verify.** It is a deterministic polynomial-time algorithm, which takes as input a tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$  where  $\text{LIST} = \{pk_1, pk_2, \dots, pk_n\}$ , a message  $m$ , and a signature  $\sigma$ , outputs a bit  $b$ . If  $b = 1$ , the signature is valid; otherwise, not.
- **Trace.** It is a deterministic polynomial-time algorithm, which takes as input a tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ , two message/signature pairs  $\{(m, \sigma), (m', \sigma')\}$ , outputs the following string: “indep”, “linked” or  $pk_i$ , where  $pk_i \in \text{LIST}$ .

**Public Traceability.** The output of the Trace algorithm looks confusing, which we discuss in detail here. For any ISSUE, any message  $m, m'$ , any  $i, i' \in [1, n]$ , we have  $(pk, sk) \leftarrow \text{KeyGen}$ ,  $\sigma \leftarrow \text{Sign}(sk, \text{TAG}, m)$ ,  $\sigma' \leftarrow \text{Sign}(sk, \text{TAG}, m)$ , it holds with an overwhelming probability that the following statement holds.

$$\text{Trace}(\text{TAG}, m, \sigma, m', \sigma') = \begin{cases} \text{“indep”}, & \text{if } i \neq i' \\ \text{“linked”}, & \text{else if } i = i', m = m' \\ pk_i, & \text{otherwise.} \end{cases}$$

**Correctness.** We define the correctness of a traceable ring signature scheme as follows. For any ISSUE, any message  $m$ , any  $i \in [1, n]$ , we have  $(pk, sk) \leftarrow \text{KeyGen}$ ,  $\sigma \leftarrow \text{Sign}(sk, \text{TAG}, m)$ , it holds with an overwhelming probability  $\text{Verify}(\text{TAG}, m, \sigma) = 1$ .

### 3.2 Constructions

In this section, we first appoint the notions that will be used.  $H_1 : \{0, 1\}^* \rightarrow G$ ,  $H_2 : \{0, 1\}^* \rightarrow G$  and  $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$  denote hash functions,  $\parallel$  denotes the concatenation of bit string,  $\perp$  denotes abort,  $\emptyset$  denotes an empty set.  $\#$  denotes the number of members in a set. We use the same elliptic curve point group  $\mathbb{G}$  of order  $q$  as the SM2 digital signature standard, where  $G$  is the generator of  $\mathbb{G}$ .

Here, we give the detailed constructions of our SM2 traceable ring signature.

1. **KeyGen.** Each user runs this algorithm, gets their key pairs.
  - (a) Each user  $U_i$  randomly chooses  $d_{A_i} \in \mathbb{Z}_q^*$  as the private key, sets the public key  $P_i = d_i \cdot G$ .
  - (b) The algorithm outputs the key pair  $(pk_i, sk_i)$ ,  $pk_i = P_i$ ,  $sk_i = d_i$ .
2. **Sign.** Given the private key  $sk_i$ , the tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ , and the message  $m$ , the user runs this algorithm to output a signature  $\sigma$ .
  - (a) The user  $A_i$  randomly chooses  $(n - 1)$  users' public keys, adds his own public key, constructs the list  $\text{LIST} = \{pk_1, pk_2, \dots, pk_n\}$ , Let  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ .
  - (b) Calculate the hash value  $W = H_1(\text{TAG})$ .
  - (c) Compute  $\sigma_i = d_i \cdot W$ .
  - (d) Compute  $A_0 = H_2(\text{TAG}, m)$  and  $A_1 = (\sigma_i / A_0)^{1/i}$ .

- (e) For all  $j \neq i$ , compute  $\sigma_j = A_0 A_1^j \in \mathbb{G}$ .
  - (f) Randomly choose  $k_i \in \mathbb{Z}_q^*$ , set  $c_{i+1} = H_3(\text{TAG}, A_0, A_1, k_i \cdot G, k_i \cdot W)$ .
  - (g) For  $j = i + 1, \dots, n, 1, \dots, i - 1$ , the user  $A_i$  chooses  $s_j \in \mathbb{Z}_q^*$ , computes  $T_j = s_j \cdot G + (s_j + c_j) \cdot P_j$ ,  $Y_j = s_j \cdot W + (s_j + c_j) \cdot \sigma_i$ , then computes  $c_{j+1} = H_3(\text{TAG}, A_0, A_1, T_j, Y_j)$ , lets  $c_1 = c_{n+1}$ .
  - (h) Compute  $s_i = \left( (1 + d_i)^{-1} \cdot (k_i - c_i \cdot d_i) \right) \bmod q$ .
  - (i) Output the traceable ring signature value  $\sigma = (A_1, c_1, s_1, s_2, \dots, s_n)$  on message  $m$ .
3. Verify. Given the tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ , the message  $m$ , and the signature  $\sigma$ , the verifier runs this algorithm to output a bit.
    - (a) If  $c_1, s_1, s_2, \dots, s_n \notin \mathbb{Z}_q^*$ , return  $\perp$ .
    - (b) For  $i = 1, 2, \dots, n$ , compute  $A_0 = H_2(\text{TAG}, m)$  and  $\sigma_i = A_0 A_1^i$ .
    - (c) Calculate the hash value  $W = H_1(\text{TAG})$ .
    - (d) For  $i = 1, 2, \dots, n$ , compute  $T_j = s_j \cdot G + (s_j + c_j) \cdot P_j$ ,  $Y_j = s_j \cdot W + (s_j + c_j) \cdot \sigma_i$ , then compute  $c_{j+1} = H_3(\text{TAG}, A_0, A_1, T_j, Y_j)$ .
    - (e) If  $c_1 = c_{n+1}$ , then outputs  $b = 1$ ; otherwise  $b = 0$ .
  4. Trace. Given the tag  $\text{TAG} = \{\text{ISSUE}, \text{LIST}\}$ , two message/signature pairs  $\{(m, \sigma), (m', \sigma')\}$ , everyone can run this algorithm.
    - (a) Calculate the hash value  $W = H_1(\text{TAG})$ .
    - (b) For  $i = 1, 2, \dots, n$ , compute  $A_0 = H_2(\text{TAG}, m)$  and  $\sigma_i = A_0 A_1^i$ .
    - (c) Similarly, for  $i = 1, 2, \dots, n$ , compute  $A_0 = H_2(\text{TAG}, m')$  and  $\sigma'_i = A_0 A_1^i$ .
    - (d) For  $i = 1, 2, \dots, n$ , if  $\sigma_i = \sigma'_i$ , store  $P_i$  on TLIST, TLIST is an empty list initially.
    - (e) Finally, perform the following steps:
      - If the public key  $P$  is the only entry in TLIST, output  $P$ ;
      - If TLIST = LIST, output “link”;
      - If TLIST =  $\emptyset$  or  $1 < \#\text{TLIST} < n$ , output “indep”.

### 3.3 Discussion

This scheme realizes the generation of traceable ring signature based on SM2 digital signature algorithm. The signer hides his identity in the signature group by collecting the users' public keys, and generates a signature label at the same time, which protects the signer's privacy, avoids the abuse of signature, and realizes the tracking of the signer by means of a secondary signature. The invention ensures the integrity, unforgeability, anonymity and traceability of the signature.

- **Integrity.** In signature phase, the signature  $\sigma = (A_1, c_1, s_1, s_2, \dots, s_n)$ , where  $A_1 = (\sigma_i/A_0)^{1/i}$  and  $A_0 = H_2(\text{TAG}, m)$ . In verification phase, the verifier should calculate the  $A_0 = H_2(\text{TAG}, m)$  to verify the signature. Consequently, the signature  $\sigma$  is generated via original data  $m$ . Once the data is tampered with, it cannot pass the verification phase, so as to ensure the integrity of data.
- **Unforgeability.** In our construction, the signature  $\sigma = (A_1, c_1, s_1, s_2, \dots, s_n)$ , where  $A_1 = (\sigma_i/A_0)^{1/i}$  and  $\sigma_i = d_i \cdot W$ . The private key  $d_i$  is only known by the signer  $U_i$ , so it is impossible for others to forge the signature without private key.

- **Anonymity.** In verification phase, the verifier uses  $\text{LIST} = \{pk_1, pk_2, \dots, pk_n\}$  to verify the signature  $\sigma$ . Besides, the auxiliary parameters  $c_1, \dots, c_n$  are generated with users' public keys rather than the signer's public key. As a result, the probability that adversary can identify the real signer is less than  $1/n$ , where  $n$  is the number of users.
- **Traceability.** In trace phase, all users in the ring should resign data  $m'$  with the same tag TAG. Recall that  $\sigma_i = A_0 A_1^i = d_i \cdot W = d_i \cdot H_1(\text{TAG})$ . If the old signature  $\sigma$  is equal to new  $\sigma'$ , it means that they are generated by the single private key  $d_i$ . Hence, we catch the signer.

## 4 Evaluation and Analysis

In this section we evaluate the performance of STRS from communication cost and computation cost.

### 4.1 Communication Cost

Let  $|q|$  be the bits of order  $q$ ,  $|\mathbb{G}|$  be the bits of group  $\mathbb{G}$ . In STRS, the Sign generates signatures  $\sigma = (A_1, c_1, s_1, s_2, \dots, s_n)$ , where  $A_1 \in \mathbb{G}$  and  $c_1, s_1, s_2, \dots, s_n \in \mathbb{Z}_q^*$ . Consequently, the communication cost of STRS are  $(n + 1)|q| + |\mathbb{G}|$ .

### 4.2 Computation Cost

This paper focus the computation cost in algorithms includes Sign, Verify and Trace. Suppose that the number of ring member is  $n$ ,  $T_{add}$  is the additive operation on group  $\mathbb{G}$ ,  $T_{mul}$  denotes the multiplication operation on group  $\mathbb{G}$ ,  $T_{exp}$  is the exponentiation operation on  $\mathbb{G}$ ,  $T_H$  is the hash function for  $\{0, 1\}^* \rightarrow G$  (e.g., the  $H_1$  and  $H_2$ ), and  $T_{H'}$  represents the hash function for  $\{0, 1\}^* \rightarrow \mathbb{Z}_q^*$  (e.g., the  $H_3$ ). In addition,  $T_{ZA}$  denotes the inverse operation on group  $\mathbb{Z}_q^*$ .

We analyze and compare STRS and other SM2-related ring signature work, the results are shown in Table 1.

**Table 1.** Computation cost for SM2 ring signature schemes.

Schemes	Sign	Verify	Trace
STRS	$2T_H + nT_{H'} + T_{ZA} + (2n - 2)T_{add} + (5n - 1)T_{mul} + (n + 1)T_{exp}$	$2T_H + (n - 1)T_{H'} + (2n - 2)T_{add} + (5n - 4)T_{mul} + nT_{exp}$	$3T_H + 2nT_{mul} + 2nT_{exp}$
SRS [13]	$nT_H + T_{ZA} + (n - 1)T_{add} + (2n - 1)T_{mul}$	$nT_H + nT_{add} + 2nT_{mul}$	–
SLRS [13]	$T_H + nT_{H'} + T_{ZA} + (2n - 2)T_{add} + (4n - 1)T_{mul}$	$nT_{H'} + T_{ZA} + 2nT_{add} + 4nT_{mul}$	–

## 5 STRS-Based Blockchain Evidence-Storage System

As a decentralized trust platform, all nodes adopt full backup mechanism to ensure availability, causing the huge storage overhead. Generally, to reduce the storage pressure of blockchain, massive data is often stored in the remote cloud, and a handful of critical proof is stored in the trusted blockchain that is used to ensure the trustworthiness of the data in the cloud. Typically, to achieve traceable evidence storage, we present SBES, a blockchain-based evidence-storage system that employs STRS to secure users' privacy while also tracing the target signer. Moreover, a traceable data structure is introduced for the traceability of evidence.

### 5.1 System Model

Figure 1 shows the system model of SBES, which involves four entities:

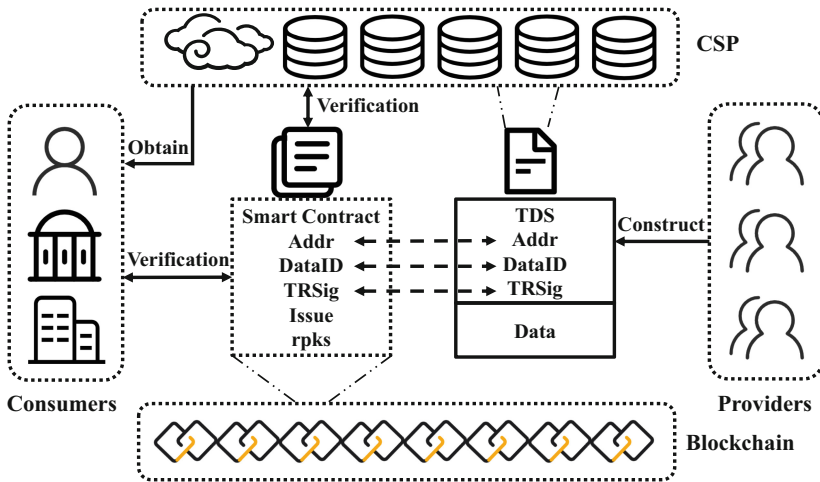


Fig. 1. System model.

- **Consumers.** They prefer to request data from cloud storage providers and evidence from blockchain for arbitration or other actions.
- **Providers.** They are in form of group to provide evidence via cloud storage providers due to limited storage capacity.
- **Blockchain.** Blockchain runs on nodes distributed all over the world, with the characteristics of decentralization, tamper-proof and supporting smart contract execution. It is accessible to everyone and responsible for storing the evidence for the data in CSP.
- **Cloud storage provider (CSP).** They are typically commercial entities, which offer users on-demand network access to a large, shared pool of storage resources.



Additionally, we design a traceable data structure for evidence-storage, named *TDS*, as shown in Fig. 1, consisting of data identity *DataID*, smart contract address *Addr*, traceable ring signature *TRSig* and data itself. Correspondingly, we also require the smart contract to trustfully record the above information, including topic *issue* (*i.e.*, the *ISSUE* in proposed scheme) and providers' public keys  $\text{rpks} = \{P_i, i \in n\}$ . Note that the data can be in the form of encryption or others, it's out of our scope.

## 5.2 Our Protocol

Our protocol is defined by a collection of phases as follow:

**Setup Phase.** In this phase,  $n$  data providers generate their cryptographic parameters. First, all providers generate their STRS parameters, *i.e.*,  $P_i$  and  $d_i$ . Next, they should generate blockchain account parameters, *i.e.*,  $pk_i, sk_i$ . Here, we suppose that the blockchain runs safely and smoothly.

**Creating Phase.** They need to construct TDS for data  $m$  before storing. First, they use the pseudo random number generator to produce a random number as the *DataID*. Second, they should generate the traceable ring signature *TRSig*. Third, providers should send transaction to deploy smart contract for the data. Finally, they fill the *Addr* to complete TDS.

**Uploading Phase.** After the provider completes the TDS, the data and its proof can be uploaded to the CSP. CSP will check the data with proof. First, CSP finds the blockchain smart contract according to the *Addr*. Next, CSP check the *DataID* and *TRSig* in smart contract whether these data are equal to those in TDS. Last, CSP uses the *rpks* to verify the *TRSig*.

**Using Phase.** The consumer who counters the desired data will send request to the CSP. After obtaining the data, the consumer can also check the data as the CSP does in **Uploading Phase**.

**Tracing Phase.** IF some data are reported illegal, CSP should delete the data while keeping and sending the trace evidence (*i.e.*, *DataID*, *Addr* and *TRSig*) to providers for accountability. Providers should execute the trace algorithm with parameters in blockchain (*i.e.*, *issue*) to disclose the malicious.

## 6 Security Analysis

In this section, we analyze the following security features of our scheme, namely: privacy-preserving, integrity, and traceability.

**Theorem 1 (Privacy-Preserving).** *For the purpose of protecting the providers' privacy, SBES is able to restrict the following behaviors of the consumers: (1) obtain the identity of provider with the off-chain data. (2) get the identity of provider with the on-chain data. (3) obtain the identity of providers while tracing.*

*Proof.* For the purpose (1), we use the traceable ring signature  $\sigma$  to replace the traditional signature that will reveal the signer. As explained in Sect. 3.3, the STRS provides anonymity for providers. The consumer can only use the public key  $\{P_i, i \in n\}$  in the group to verify the signature. Hence, the consumer cannot judge which provider signed the data. For the purpose (2), the blockchain, such as Bitcoin [20] and Ethereum [21], provides pseudonyms to protect identity. Although some studies [22, 23] have been proposed to realize de-anonymity, the success rate and accuracy rate are still very low. For the purpose (3), as shown in the Trace algorithm, only the provider whose  $\sigma'_i = \sigma$  will be pushed into **TList** and be revealed. The rest providers are still protected by SBES.  $\square$

**Theorem 2 (Integrity).** *SBES is able to ensure the data integrity in the evidence-storage process.*

*Proof.* As explained in Sect. 3.3, the STRS provides integrity for providers with signature  $\sigma$ . The signature  $\sigma$  is generated via original data  $m$ . Once the data  $m$  is replaced by  $m'$ , the  $A_0 = H_2(\text{TAG}, m')$ , resulting the  $c_1 \neq c_{n+1}$ . As a result, it cannot pass the verification phase, so as to ensure the integrity of data.

**Theorem 3 (Traceability).** *In SBES, any provider who violates the protocol would be exposed.*

*Proof.* As the blockchain is maintained by all nodes via the consensus mechanism, the data in blockchain smart contract is reliable. When one provider violates the protocol, the group can execute the Trace algorithm with the data in blockchain (*i.e.*, **issue**). They first compute the hash  $W = H_2(\text{TAG})$  with **issue**. Second, all providers compute the  $\sigma'_i = A_0 A_1^i$ , where  $A_0 = H_2(\text{TAG}, m')$ ,  $m'$  is a string of arbitrary length. Third, if  $\sigma_i = \sigma'_i$ , the public key  $P_i$  is pushed into **TList**. Finally, the **TList** records all the malicious providers.

## 7 Conclusion

Traceable ring signature (TRS) scheme has a wide range of applications, such as an evidence-storage system. On the one hand, it has the complete anonymity and unforgeability of ring signatures. On the other hand, it effectively avoids the problem of regulation faced by traditional ring signatures. Researchers have proposed many TRS algorithms with different forms and characteristics, but there is no TRS based on SM2 digital signature algorithm. To promote the application of SM2 digital signature algorithm in these fields, a traceable ring signature based on SM2 digital signature algorithm, named STRS, is proposed in this paper. Furthermore, we describe an STRS-based blockchain evidence-storage system (SBES) in which users submit evidence with a signature generated by STRS, and the regulator can figure out the identity of the signer.

**Acknowledgments.** This work was supported in part by the Finance Science and Technology Project of Hainan Province (No. ZDKJ2020009); in part by the National Key R&D Program of China (No. 2021YFB2700601); in part by the National Natural Science Foundation of China (Nos. 62163011, 62072092, 62072093 and U1708262); in part by the Fundamental Research Funds for the Central Universities (No. N2023020); in part by the Natural Science Foundation of Hebei Province (No. F2020501013); in part by the China Postdoctoral Science Foundation (No. 2019 M653568); and in part by the Key Research and Development Project of Hebei Province (No. 20310702D).

## References

1. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-45682-1\\_32](https://doi.org/10.1007/3-540-45682-1_32)
2. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). [https://doi.org/10.1007/3-540-46416-6\\_22](https://doi.org/10.1007/3-540-46416-6_22)
3. Sun, S.-F., Au, M.H., Liu, J.K., Yuen, T.H.: RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero. In: Foley, S.N., Gollmann, D., Sneekenes, E. (eds.) ESORICS 2017. LNCS, vol. 10493, pp. 456–474. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-66399-9\\_25](https://doi.org/10.1007/978-3-319-66399-9_25)
4. Yuen, T.H., et al.: RingCT 3.0 for blockchain confidential transaction: shorter size and stronger security. In: Bonneau, J., Heninger, N. (eds.) FC 2020. LNCS, vol. 12059, pp. 464–483. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-51280-4\\_25](https://doi.org/10.1007/978-3-030-51280-4_25)
5. Zhang, F., Huang, N.N., Gao, S.: Privacy data authentication schemes based on Borromean ring signature. *J. Cryptol. Res.* **5**(5), 529–537 (2018). <https://doi.org/10.13868/j.cnki.jcr.000262>
6. Monero: About Monero. <https://getmonero.org/knowledge-base/about>. Accessed 4 Mar 2022
7. Kolachala, K., et al.: SoK: money laundering in cryptocurrencies. In: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, pp. 5:1–5:10 (2021). <https://doi.org/10.1145/3465481.3465774>
8. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-71677-8\\_13](https://doi.org/10.1007/978-3-540-71677-8_13)
9. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-27800-9\\_28](https://doi.org/10.1007/978-3-540-27800-9_28)
10. Fujisaki, E.: Sub-linear size traceable ring signatures without random oracles. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 393–415. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-19074-2\\_25](https://doi.org/10.1007/978-3-642-19074-2_25)
11. Au, M.H., et al.: Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction. *Theor. Comput. Sci.* **469**, 1–14 (2013). <https://doi.org/10.1016/j.tcs.2012.10.031>
12. Scafuro, A., Zhang, B.: One-time traceable ring signatures. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021. LNCS, vol. 12973, pp. 481–500. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-88428-4\\_24](https://doi.org/10.1007/978-3-030-88428-4_24)

13. Fan, Q., He, D.B., Luo, M., Huang, X.Y., Li, D.W.: Ring signature schemes based on SM2 digital signature algorithm. *J. Cryptol. Res.* **8**(4), 710–723 (2021). <https://doi.org/10.13868/j.cnki.jcr.000472>
14. Peng, C., He, D.B., Luo, M., Huang, X.Y., Li, D.W.: An identity-based ring signature scheme for SM9 algorithm. *J. Cryptol. Res.* **8**(4), 724–734 (2021). <https://doi.org/10.13868/j.cnki.jcr.000473>
15. State Cryptography Administration: Public key cryptographic algorithm SM2 based on elliptic curves - Part 2: digital signature algorithm (2010). <http://www.sca.gov.cn/sca/xwdt/2010-12/17/1002386/files/b791a9f908bb4803875ab6aeeb7b4e03.pdf>
16. Su, Z., et al.: LVBS: lightweight vehicular blockchain for secure data sharing in disaster rescue. *IEEE Trans. Depend. Secur. Comput.* **19**(1), 19–32 (2020). <https://doi.org/10.1109/TDSC.2020.2980255>
17. Li, T., et al.: Synchronized provable data possession based on blockchain for digital twin. *IEEE Trans. Inf. Forensics Secur.* **17**, 472–485 (2022). <https://doi.org/10.1109/TIFS.2022.3144869>
18. Cui, L., et al.: A blockchain-based containerized edge computing platform for the internet of vehicles. *IEEE Internet Things J.* **8**(4), 2395–2408 (2021). <https://doi.org/10.1109/JIOT.2020.3027700>
19. Kircanski, A., Shen, Y., Wang, G., Youssef, A.M.: Boomerang and slide-rotational analysis of the SM3 hash function. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 304–320. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-35999-6\\_20](https://doi.org/10.1007/978-3-642-35999-6_20)
20. Nakamoto S.: Bitcoin: a peer-to-peer electronic cash system. *Decent. Bus. Rev.* 21260 (2008). <https://www.debr.io/article/21260.pdf>
21. Wood, G., et al.: Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151(2014), pp. 1–32 (2014). <https://files.gitter.im/ethereum/yellowpaper/VIyt/Paper.pdf>
22. Kappos, G., et al.: An empirical analysis of anonymity in Zcash. In: 27th USENIX Security Symposium (USENIX Security 2018), Baltimore, MD, USA, pp. 463–477 (2018). <https://www.usenix.org/conference/usenixsecurity18/presentation/kappos>
23. Biryukov, A., Tikhomirov, S.: Deanonymization and linkability of cryptocurrency transactions based on network analysis. In: 2019 IEEE European symposium on security and privacy (EuroS&P), Stockholm, Sweden, pp. 172–184 (2019). <https://doi.org/10.1109/EuroSP.2019.00022>