On The Impossibility of Providing Strong Anonymity in Blockchains via Linkable Ring Signatures

Huang Zhang[1], *Fangguo Zhang*[2,3*], *Ke Gu*[1]

[1] *School of Computer and Communication Engineering, Changsha University of Science and Technology, China*
[2] *School of Computer Science and Engineering, Sun Yat-sen University, China*
[3] *Guangdong Province Key Laboratory of Information Security Technology, China*
learnwitherror@gmail.com, isszhfg@mail.sysu.edu.cn, gk4572@163.com

Abstract

Anonymity is a necessary property for a ring signature scheme and also its variant such as linkable ring signature and traceable ring signature schemes, which are especially useful in blockchains. Intuitively, those variants were designed for detecting or seeking the dishonest signatory, however, at the cost of reducing the anonymity of a traditional ring signature. As a result, while various constructions of strongly anonymous ring signatures were well-known, a linkable ring signature scheme with the same property was an open problem for a long time.

In this work, we launched a so-called denying attack to show the gap between an arbitrary ring signature and linkable ring signature transparently, which further confirmed the widely believed impossibility in building a linkable ring signature with both strong anonymity and strong linkability.

For a concrete instance, we also applied this attack to the scheme in IEEE TKDE, which to the best of our knowledge is the unique linkable ring signature both with strong anonymity and strong linkability so far.

The concrete attack is easily launched in blockchain so that it shows the impossibility of providing strong anonymity via linkable ring signature for blockchain applications, since strong likability is indispensable.

Keywords: Blockchain, Denying attack, Linkable ring signature, Strong anonymity, Strong linkability

1 Introduction

Anonymity is a long-standing subject that cryptographers concerned for the clients of computer or network systems. To this end, digital signatures were modified to ring signature [20] and group signature [2-3] schemes for distinct requirements. When only considering anonymity, a ring signature is especially useful since it could support this property against unbounded adversaries and it typically enjoys lower computation costs.

With the rapidly increasing passion to blockchain technology [13-14, 24-25, 28], users also regard anonymity as one of the crucial properties that whether a blockchain-based system could afford. Originated from the anonymity concerns to block-chain-based cryptocurrencies [18, 21], ring signatures gain amounts of developments even in the post-quantum scenario [5-6, 9, 27]. For clients who are willing to have better efficiency at the cost of relatively weaker anonymity, CryptoNote and Monero suggested a ring-signature-based cryptocurrency, instead of fully NIZK-based system [12, 16, 23], and leave the task of amounts hiding to the technique of confidential transactions [15].

Nevertheless, a ring signature should be transformed into a linkable ring signature [11] in a cryptocurrency system, so that double spending could at least be detected without the help of any secret information. In this case and also in many e-voting systems, the linkability of linkable ring signatures is crucial. Through this fact, it is nature to consider that the linkability certainly reduces the anonymity of a ring signature, and hence a linkable ring signature cannot reach the strongest level anonymity.

In a ring signature scheme, strong anonymity (or unconditional anonymity) is the highest requirement, since it is a desirable property without any computational assumptions. Fortunately, A large number of ring signature schemes provide strong anonymity, *e.g.*, [1, 20, 26]. But as discussed before, when designing a linkable ring signature with strong anonymity, intractability rises. Liu *et al*. left the construction of such a scheme as an open problem [11]. Short after that, Jeong *et al*. stated and proved that it is impossible to make a ring signature scheme that provides strong anonymity together with strong linkability [8]. The very different conclusion was introduced by Liu *et al*., which interpreted strong anonymity into two distinct definitions (see Sect. 2.3 for details) and designed a linkable ring signature according to the slightly weaker definition [10].

The skills in [10] are elegant, in the sense of a totally different public key generation strategy. Actually, the corresponding private key is perfectly hiding by the public key (Pedersen commitment), so that no unbounded adversary can extract the correct private key and further determine the real signer by the slightly weaker definition. However, we find that the existence of such a scheme is not able to remove the gap of anonymity between linkable and conventional ring signatures, and in the practical point of view, the slightly weaker definition of strong anonymity has limited advantages comparing to the weak anonymity of linkable ring signatures. Consequently, in this paper, we presented a general attack,

*Corresponding Author: Fangguo Zhang; E-mail: isszhfg@mail.sysu.edu.cn
DOI: 10.53106/160792642023032402028

called denying attack, on arbitrary linkable ring signature schemes. With this attack, the distance from a ring signature to a linkable ring signature was shown transparently, which further confirmed the impossibility result proposed by Jeong *et al*. [8]. We also give discussions on the cost of an adversary to implement a concrete attack to the scheme in [10]. The cost is relatively low even such a scheme is deployed in a blockchain-based system.

The remaining of this paper is organized as follows. In Sect. 2, we give some conventions and definitions employed in the whole paper. Section 3 recalls the contradiction between strong linkability and strong anonymity of a linkable ring signature scheme, together with the scheme that is opposed to this contradiction. We propose our general attack on arbitrary linkable ring signatures in Sect. 4. A concrete attack to the scheme in [10] is given in Sect. 5. At last, we make a brief conclusion in Sect. 6.

2 Preliminaries

2.1 Notations

We use **Z**, **N** to denote the set of all integers and natural numbers, and PPT to indicate probabilistic polynomial time. If A is a probabilistic algorithm, then $a \leftarrow A(\mathrm{x})$ denotes that this algorithm outputs a random variable a on input x with its own coin-tossing. If S is a finite set, $a \leftarrow S$ indicates that a is sampled from S uniformly at random. When a, b are two integers such that $a \le b$, $[a, b]$ is the set $\{x : a \le x \le b\}$. Moreover, for notation simplicity, we will use $\{x_i\}_{i=a}^{b}$ to denote the set $\{x_a, x_{a+1}, ..., x_b\}$, and a user of a signature scheme will be named by his/her public key. If a function f : $\mathbf{N} \rightarrow \mathbf{N}$ vanishes faster than the reciprocal of any positive polynomial function poly : $\mathbf{N} \rightarrow \mathbf{N}$, we say f is negligible, written $f(n) = \mathrm{negl}(n)$.

2.2 Linkable Ring Signature

A linkable ring signature is a tuple of PPT algorithms (**Setup**, **KGen**, **Sign**, **Verify**, **Link**).

k_i •$param \leftarrow$ **Setup**(1^λ): On input a security parameter λ, the algorithm generates and publishes the system parameters $param$. $params$ will be the default input of the other algorithms. Denote by EID, M the domains of event-id and messages, respectively.

•(pk, sk) ← **KGen**($param$): This algorithm generates a public and private key pair (pk, sk).

•$\sigma \leftarrow$ **Sign**($event$, n, Y, sk, M): Output a signature σ on the message $M \in \mathrm{M}$ with respect to the ring Y and the event-id $event \in$ EID. It is required that the public key pk corresponding to sk is in Y.

•**accept**/**reject** ← **Verify**($event$, n, Y, M, σ): Verify a purported signature σ on a message M with respect to the ring Y and the event-id $event$. It outputs **accept** if accepting and **reject** if rejecting the signature.

• **link**/**unlink** ← **Link**($event$, n_1, n_2, Y_1, Y_2, M_1, M_2, σ_1, σ_2): On input two accepting signatures σ_1, σ_2 on the same event-id $event$, output **link** if the signatures are linked, and output **unlink** otherwise.

2.3 Strong Anonymity

In history, the strong anonymity was lack of a rigorous security model, and the following is a definition summarized from [4] and [20].

Definition 1 (Introduction, [8]). *Any party cannot know the actual signer of a ring signature, even if all of the private keys of the parties of the ring are known.*

According to the above situation, Liu *et al*. found some subtleties in the above definition and further interpreted it into two distinctive ways. The first one is slightly stronger.

Definition 2 (Definition 1, [10], informal). *Given the public keys (or identities in the case of ID-based scheme) of all parties of a ring signature, any party cannot know the actual signer even if all of the private keys* ***owned by*** *the parties of the ring are known.*

In the remaining of the paper, we will call the private key owned by a signer the actual private key. The second definition is somewhat weaker than the former.

Definition 3 (Definition 2, [10], informal). *Given the public keys (or identities in the case of ID-based scheme) of all parties of a ring signature, any party cannot know the actual signer even if all of the private keys* ***corresponding to*** *the parties of the ring are known.*

The subtleties were hidden in the phrases marked in bold. For a (linkable) ring signature scheme, if the map from the private key space to the public key space is a bijection, then the above two definitions are equivalent. But if the foregoing condition is not satisfied (*e.g.*, surjection), these two definitions yield completely different feedbacks, since in Definition 3, an adversary is not able to extract the actual private key of a real signatory because of information loss. We shall see the details in Sect. 3.

To prevent ambiguities, we also give the formal definitions for Definition 2 and 3 on linkable ring signature below. As anonymity in Definition 2 is originated from ring signatures and described the strongest notion, we directly use the definition in [7] with slight modifications to adapt linkable ring signartures.

Definition 4 (Definition 12, [7]). *A linkable ring signature scheme* (**Setup**, **KGen**, **Sign**, **Verify**, **Link**) *has strong anonymity*, *if a signature on a message M and an event-id event, under a ring Y*, *and public key* pk_0 *looks exactly the same as a signature on the message M and the event-id event, under the ring Y and public key* pk_1. *This means that the signer's key is hidden among all the honestly generated keys in the ring*. *Formally*, *we require that for any* (*unbounded*) *adversary A*

$$\Pr\left[\begin{array}{c} param \leftarrow \mathbf{Setup}(1^\lambda); \\ (M, i_0, i_1, Y, event) \leftarrow A^{\mathbf{KGen}}(\cdot) \\ b \leftarrow \{0,1\}; \\ \sigma \leftarrow \mathbf{Sign}(event, n, Y, sk_{i_b}, M) \end{array} : A(\sigma) = b \right] = \frac{1}{2} \quad \textbf{(1)}$$

where A chooses i_0, i_1 such that (pk_{i0}, sk_{i0}), (pk_{i1}, sk_{i1}) have been generated by the key generation oracle **KGen**(·) and $pk_{i_0}, pk_{i_1} \in Y$.

To formally define the strong anonymity of linkable ring signature schemes regarding to Definition 3, [3] stated that

the following oracles should be considered first.

• $pk_i \leftarrow \mathrm{JO}(\perp)$. The *Joining Oracle*, on request adds a new user to the system. It returns the public key *pk* of the new user.

• $sk_i \leftarrow \mathrm{CO}(pk_i)$. The *Corruption Oracle*, on input a public key pk_i that is query output of JO, returns the corresponding (possibly not the actual) private key sk_i.

• $\sigma' \leftarrow \mathrm{SO}(event, n, Y, pk_\pi, M)$. The *Signing Oracle*, on input an event-id *event*, a group size *n*, a set *Y* of *n* public keys, the public key of the signer $pk_\pi \in Y$, and a message *M*, returns valid signature σ'.

If the scheme is proven in the random oracle model, a random oracle is simulated.

Strong anonymity in definition 3 is defined in the following game between the Simulator *S*, and the unbounded Adversary *A* is given access to oracle JO:

1. *S* generates and gives *A* the system parameters *param*.
2. *A* may query JO according to any adaptive strategy.
3. *A* gives *S* an event-id *event* ∈ EID, a group size *n* ∈ **N**, a set *Y* of *n* public keys such that all of the public keys in *Y* are query outputs of JO, and a message *M* ∈ M. Parse the set *Y* as $\{pk_1, \ldots, pk_n\}$. *S* randomly picks $\pi_R \in \{1, \ldots, n\}$ and computes $\sigma_\pi \leftarrow$ **Sign***(event, n, Y, sk_π, M)*, where sk_π is a corresponding private key of pk_π, σ_π is given to *A*.
4. *A* outputs a guess $\pi' \in \{1, \ldots, n\}$.

We denote the advantage of *A* in winning this game by

$$\mathrm{Adv}_A^{\mathrm{Anon}}(\lambda) = \left| \Pr[\pi' = \pi] - \frac{1}{n} \right|. \tag{2}$$

Definition 5 (Definition 4, [3]). *A linkable ring signature scheme is strongly anonymous if for any unbounded adversary A,* $\mathrm{Adv}_A^{\mathrm{Anon}}(\lambda)$ *is zero.*

2.4 Strong Linkability

Informally, linkability ensures that any two signatures generated using the same private key and event-id *event* (see Sect. 2.2 for parameter descriptions) will be determined by all system users. The following definition is not fully extracted from [11], since some of the statements are now refined as nonslanderability.

Definition 4 (Definition 3, [11]). *Let* Y_1, Y_2 *be two lists of* n_1 *and* n_2 *public keys. A linkable ring signature scheme with unforgeability is strong linkable if there exists a* PPT *algorithm* F_1 *which outputs* **link**/**unlink** *with probability*

$$\Pr\left[\begin{matrix} F_1(event, n_1, n_2, Y_1, Y_2, M_1, M_2, \\ \sigma_1, \sigma_2) = \textbf{unlink} \mid sk_{\pi 1} \neq sk_{\pi 2} \end{matrix} \right] \leq \mathrm{negl}(\lambda). \tag{3}$$

for all sufficiently large λ, *any* $\pi_1 \in [1, n_1]$, $\pi_2 \in [1, n_2]$, *any messages* M_1, M_2 *and* $\sigma_1 \leftarrow$ **Sign**(*event*, n_1, *Y*, $sk_{\pi 1}$, M_1), $\sigma_2 \leftarrow$ **Sign**(*event*, n_2, Y_2, $sk_{\pi 2}$, M_2).

The above definition is also called strong linkability, and the algorithm F_1 is actually the algorithm **Link** in a linkable ring signature. The phrase "strong" means that the requirement to have a signer to be linked is mandatory which is opposed to weak linkability introduced in [8].

3 A Discussion on Strong Linkability and Anonymity

In this section, we introduce the reason why it is hard to design a linkable ring signature with both strong anonymity and strong linkability. Subsequently, we describe the key idea for [10] to construct a scheme that is opposed to the obstacle.

3.1 The Claim on Impossibility

Generally speaking, in a linkable ring signature scheme, if a signer wishes to sign a message, he/she has to embedded the information of his/her private key into the ingredient for linking (*e.g.*, the parameter *t* in Sect. 3.2) so that the signature could pass the verification algorithm. Furthermore, the embedded information of the private key is responsible for strong linkability of the scheme.

Consequently, if we take Definition 2 into consideration, in which an adversary is able to obtain all private keys owned by the ring members of a signature, then the correctness of the following theorem is obvious.

Theorem 1 (Theorem 1, [8]). *It is impossible to make a linkable ring signature scheme that provides both strong anonymity (interpreted as in Definition 2) and strong linkability.*

The proof in [8] is simple, and the main idea is that, given a target linkable ring signature, a PPT adversary could grasp all the private keys of the ring members, and then signs arbitrary messages using the same event-id and the private keys one after another. Finally, the algorithm **Link** would help the adversary to determine which of the members is the actual signer of the target signature, since the scheme is strongly linkable.

However, the work of [10] showed that if the strong anonymity is interpreted as in Definition 3, we would obtain a completely different resulting conclusion.

3.2 Linkable Ring Signature with Strong Anonymity and Likability

This section is an elegant work proposed in [10], which supports both strong linkalibility and strong anonymity. We should notice that the anonymity of the scheme in [10] is only satisfied Definition 3.

The linkable ring signature consists of a tuple of five PPT algorithms (**Setup**, **KGen**, **Sign**, **Verify**, **Link**).

•**Setup**: On input a security parameter, this algorithm generates a cyclic group *G* of prime order *p* such that the underlying discrete logarithm problem is intractable. Let $H: \{0,1\}^* \rightarrow G$ and $H': \{0,1\}^* \rightarrow \mathbf{Z}_p$ be two cryptographic hash functions. Let $g = H$("GENERATOR-g") and $h = H$("GENERATOR-h"). Return *param* = (*G*, *g*, *h*, *p*, *H*, *H'*, "GENERATOR-g", "GENERATOR-h") as the system parameters.

• **KGen**: A user randomly chooses $x, y \leftarrow \mathbf{Z}_p$ and computes $Z = g^x h^y$. His/Her private key is $sk = (x, y)$ and the corresponding public key is $pk = Z$.

• **Sign**: On input (*event*, *n*, *Y*, sk_π, *M*), where *event* is the event-id, *n* is the number of members included in the ring signature, $Y = \{pk_1, ..., pk_n\} = \{Z_1, ..., Z_n\}$ is the set of public keys of members in the ring, sk_π is the private key corresponding to the public key pk_π such that $pk_\pi \in Y$ (*w.l.o.g.*, $\pi \in [1,n]$), and *M* is the message to be signed, the member (with the knowledge of $sk_\pi = (x, y)$) computes the following:

- Compute $e = H(event)$ and $t = e^x$.
- Randomly generate $r_x, r_y, c_1, ..., c_{\pi-1}, c_{\pi+1}, ..., c_n \leftarrow \mathbf{Z}_p$ and compute $K = g^{r_x} h^{r_y} \prod_{i=1, i\neq\pi}^{n} Z_i^{c_i}$, $K' = e^{r_x} t^{\sum_{i=1, i\neq\pi}^{n} c_i}$.
- Find c_π such that
  $c_1 + \cdots + c_n \bmod p = H'(Y||event||t||M||K||K')$.
- Compute
$$\overline{x} = r_x - c_\pi x \bmod p$$
$$\overline{y} = r_y - c_\pi y \bmod p$$
- Output the signature $\sigma = (t, \overline{x}, \overline{y}, c_1, ..., c_n)$.

• **Verify**: On input (*event*, *n*, *Y*, *M*, σ), first compute $e = H(event)$ and $c_0 = H'(Y||event||t||M||g^{\tilde{x}} h^{\tilde{y}} \prod_{i=1}^{n} Z_i^{c_i} || e^{\tilde{x}} t^{\sum_{i=1}^{n} c_i})$ then check whether $\sum_{i=1}^{n} c_i \bmod p = c_0$. Output **accept** if the above equation holds. Otherwise, output **reject**.

• **Link**: On input two signatures $\sigma_1 = (t_1, \cdot)$, $\sigma_2 = (t_2, \cdot)$, two messages M_1, M_2, and an event-id *event*, first check whether two signatures are valid. If yes, output **link** if $t_1 = t_2$ and output **unlink** otherwise.

In this scheme, a public key is essentially a Pedersen commitment [19] and the corresponding private key can be regarded as one of its openings. Since the Pedersen commitment is perfectly hiding, even if an adversary is computationally unbounded, the advantage for it to figure out the actual private keys of the members of a target signature is obviously negligible.

Without the ability to catch the signer's actual private key, the attack implied in Theorem 1 is hard to be implemented. This is an expected result, but we shall see in the next section that, a well-designed ring signature could also satisfy Definition 2. Consequently, it is nature to think that the gap between Definition 3 and Definition 2 is nonnegligible. In other words, an adversary of linkable ring signature schemes in Definition 3 is not strong enough, even it can be computationally unbounded.

4 General Attack

We have discussed in Sect. 3 that only under Definition 3, the specific linkable ring signature scheme is of strong anonymity. The key point is the assumption that an adversary cannot obtain the actual private keys of the ring members. However, a strongly anonymous ring signature could also satisfy Definition 2.

In our point of view, an adversary is an abstract concept, so that it could deploy much more strategies in a black-box manner to break the anonymity of the real signer, rather than just using its computational resources to solve an information theoretically infeasible problem (*i.e.*, breaking the hiding property of the Pedersen commitment). As a result, the major idea in denying attack is to extract the actual private keys of a target signature in a practically feasible manner, so as to break the assumption in Definition 3 (phrase "**corresponding to**").

The following are two possible approaches for adversaries to obtain the actual private keys of the members of a target signature. But we will mainly consider the latter one in this paper, since it is easy for an adversary in Case 1 to launch an attack described in Case 2.

Without loss of generality, letting $\{(pk_i, sk_i)\}_{i=1}^{n}$ be a set of key pairs of a linkable ring signature with strong linkability. Let (M, σ_1) be a message-signature pair generated by using private key sk_1, ring $Y = \{pk_i\}_{i=1}^{n}$, and event-id *event*, an adversary could obtain the actual private keys of the members of *Y* in two ways.

1. An adversary could force the members of Y to publish their actual private keys, $sk_1, ..., sk_n$. For $i \in [1,n]$, if the member who holds pk_i could deliver $sk_i' \neq sk_i$ to cheat the adversary that sk_i' is the private key owned by pk_i, then the member could generate two signatures $\sigma_i \leftarrow$ **Sign**(*event*, *n*, *Y*, sk_i, *M*), and $\sigma_i' \leftarrow$ **Sign**$(event, n, Y, sk_i', M)$ such that σ_i and σ_i' are not linked. This would yield a contraction to the fact that this scheme is strongly linkable. As a concrete instance, if the scheme is the one in Sect. 3, since the Pedersen commitment is computationally binding, a resource-bounded member can hardly open his/her public key in different ways. With the actual private keys, the adversary is easy to determine as in Theorem 1 that pk_1 is the owner of σ_1.

2. The members, $pk_2, ..., pk_n$ are aware of the fact that σ_1 was not signed by them, so that they could show to anyone the following statements.

(a) They can sign on behalf of *Y* with the same event-id and obtain the signatures $\{\sigma_i\}_{i=2}^{n}$. This step shows that $\{\sigma_i\}_{i=2}^{n}$ were signed by the members of *Y*, because of the guarantees given by unforgeability;

(b) $\{\sigma_i\}_{i=2}^{n}$ are not linked to σ_1. Since the scheme is strongly linkable, the signers of $\{\sigma_i\}_{i=2}^{n}$ are not the one generating σ_1;

(c) For $i \in [2,n]$, state that pk_i is the actual signer of σ_i. One possible approach to prove this statement is to sign a useless message with a ring of cardinality 1 (or a ring with only arbitrary number of duplicated members).

If someone is convinced by the above three statements, then he/she could confirm that pk_1 is the actual signer of σ_1.

Notice that, for a ring signature scheme that satisfies

Definition 2, neither of the two situations cause problems. For instance, in Case 2, step 2b is infeasible to be implemented since there is no strong linkable guarantees in a conventional ring signature scheme.

The strategy of the attack in Case 2 is essentially that the members of σ_1 except for the actual signer deny that σ_1 was signed by them. Thus, we call it the denying attack.

5 Concrete Attack and Its Costs

This section is a concrete denying attack on the scheme in Sect. 3.2. The attack is easily launched in e-voting and blockchain applications such as cryptocurrencies by directly publishing the corresponding signatures. Let us start with an observation.

Observation 1. *In the linkable ring signature scheme described in Sect. 3.2, the linking parameter t in a signature is only depending on its event-id and its corresponding actual private key sk* = (*x*, *y*).

With the observation, we know that if a user runs the signing algorithm twice with the same event-id and private key, for instance,

$\sigma = \mathbf{Sign}(event, n, Y, sk, M)$,

$\sigma' = \mathbf{Sign}(event, n', Y', sk, M')$,

then

$\mathbf{Link}(event, n, n', Y, Y', M, M', \sigma, \sigma') = \mathbf{Link}$,

even though $n \neq n'$, $Y \neq Y'$, $M \neq M'$. As a result, Step 2a and Step 2c in the general attack could be made simultaneously in the concrete attack. But if the event-id *event* is the description of the ring (*i.e., Y* or *Y′*), then these steps should be done in order, otherwise Step 2b is not convincible.

5.1 The Attack

Let $\{(pk_i, sk_i)\}_{i=1}^{n}$ be *n* public-private key pairs of the linkable ring signature scheme, where for $i \in [1, n]$, $pk_i = g^{x_i} h^{y_i}$ and $sk_i = (x_i, y_i)$ is the private key owned by pk_i.

Without loss of generality, assume that $Y = \{pk_i\}_{i=1}^{n}$ be the list of public keys and the user who holds (pk_1, sk_1) runs **Sign**(*event*, *n*, Y_1, sk_1, *M*) to generate a target signature $\sigma_1 = (t_1, \overline{x}_1, \overline{y}_1, c_1^{(1)}, \ldots, c_n^{(1)})$ for message *M*. Depending on the anonymity of the scheme, no one would know the actual signer is pk_1 when he/she only sees σ_1, *event*, *n*, *Y*, and *M*. However, the members pk_2, ..., pk_n are aware of that σ_1 does not belong to them. If they wish, they could do as in the general attack to expose the actual signer pk_1.

For $i \in [2, n]$, the user who holds pk_i runs **Sign**(*event*, 1, pk_i, sk_i, *M*) to generate a linkable ring signature $\sigma_i = (t_i, \overline{x}_i, \overline{y}_i, c_1^{(i)}, \ldots, c_n^{(i)})$ with the same event-id *event* as in σ_1 and only one ring member. If **Verify**(*event*, 1, pk_i, *M*, σ_i) = **accept**, everyone would be convinced that σ_i was signed by pk_i like what have been done in Step 2c in the general attack. Moreover, since pk_i was involved in Y_1, it is obvious that the signer of σ_i is a member of σ_1, and this is what should be proved in Step 2a in the general attack.

Finally, for $i \in [2, n]$, if **Link**(*event*, *n*, 1, *Y*, pk_i, *M*, *M*, σ_1, σ_i) = **unlink,** then everyone could confirm that (pk_i, sk_i) is not the key pair signed σ_1 before, depending on Observation 1. This is the statement that should be shown in Step 2b in the general attack. With all the above facts, all users could know pk_1 is the actual signer of σ_1.

5.2 The Costs

Somebody will suspect that this attack would damage to the members' benefits as well, since they should leak information about their private keys. However, it can hardly ensure that the members will stop their attack just because of the costs. Such a conclusion comes from a point of view somewhat out of the region of cryptography or algorithm. It is depending on whether the benefit from an attack for the members is greater than the costs. Moreover, in some situations, the costs are indeed relatively low or can be reduced by attackers. Let us consider the motivations of the members to launch such an attack in two distinct cases.

The first case is according to the observation that the current denying attack typically does not put new useful signatures of the attackers at a risky stage. In this case, *e.g.*, e-voting system, the attackers can still generate fresh unlinked signature with new event-id.

Let {(pk_1, sk_1), ..., (pk_n, sk_n), ..., (pk_m, sk_m)} be *m* public-private key pairs of the linkable ring signature scheme, where for $i \in [1, m]$, $pk_i = g^{x_i} h^{y_i}$ and $sk_i = (x_i, y_i)$ is the private key owned by pk_i. We further assume that $Y_1 = \{pk_i\}_{i=1}^{n}$, and $Y_2 = \{pk_i\}_{i=n}^{m}$ be two sets of public keys, where *event*, (pk_1, sk_1), Y_1, *M*, $\sigma_1 = (t_1, \cdot)$ is the event-id, the key pair, the ring, the message and the signature that were attacked.

Even though for $i \in [2, n]$, the *i*-th user has to publish $\sigma_i = (t_i, \cdot)$, where $t_i = H(event)^{x_i}$, to implement the denying attack, there is no negative influence for he/she to sign another message if his/her event-id in the new signature is that $event' \neq event$.

For instance, assume that the event-id of a signature in the scheme is the description of the members of the corresponding ring, and the user who holds (pk_n, sk_n) signs his/her message M' on behalf of Y_2 (*i.e.*, *event* = Y_2). The resulting signature would be $\sigma' = (t', \overline{x}, \overline{y}, c_1, \ldots, c_n)$. According to the collision-resistance of the hash function

$$\Pr\left[\begin{array}{l} t' = H(event')^{x_n} = H(Y_2) \neq \\ H(Y_1) = H(event)^{x_n} = t_n \end{array}\right] \geq 1 - \mathrm{negl}(\cdot) \qquad (4)$$

where $t_n \in \sigma_n$, and $\sigma_n \leftarrow \mathbf{Sign}(event, 1, pk_n, sk_n, M)$ is the published signature of pk_n in the denying attack.

Relying on the algorithm **Link**, the new generated signature σ' will not be linked to σ_n generated by the *n*-th member in the foregoing attack with high probability, except for that σ' is still vulnerable under the denying attack to the event-id $event'$ by the members in Y_2. However, the risk is essentially introduced by the impossibility of a linkable ring

signature. Hence, in the positive point of view, the costs for the user who holds (pk_n, sk_n) to join the denying attack is acceptable, since they can still generate unlinked signature in a new event if nobody denying attacks them later. On the opposite side, the "Prisoner's dilemma" also gives us a negative viewpoint. That is, as anybody could be denying attacked, the users of a linkable-ring-signature-based system may perhaps choose to attack the others first. Consequently, if an application system cannot break the conditions for launching a denying attack, the users indeed have their motivations to break the anonymity of the system, even though they will lose their anonymity either.

The second is corresponding to blockchain-based applications, especially the cryptocurrency system. If the attack is launched to cryptocurrency, such as Monero and CryptoNote, the situation is slightly different. Concretely, in a ring-signature-based cryptocurrency system, the signature scheme should be modified to one-time signature to prevent double-spending, so that a private key can only sign one message during the whole life of the system (see [22]). Consequently, if a linkable ring signature is adopted in such a blockchain system, the parameter event-id should be a predetermined constant. In other words, the discussion in the first case about that the attacker can still sign a message on a distinct event-id with limited costs is not possible in this case. Similarly, as we discussed before, this can perhaps hardly remove the motivation of attackers. Additionally, a key pair of linkable ring signature scheme is not tightly associated with a user in a ring-signature-based cryptocurrency system, since such a system introduced untraceability and unlinkability, and provided several strategies to support them [22]. As a result, the impossibility of linkable ring signatures only affects the untraceability of a cryptocurrency. In other words, denying attackers in such a system possibly could reduce their costs. We proceed to give an experimental approach.

Let $\{(pk_i, sk_i)\}_{i=1}^{n}$ be n public-private key pairs of the linkable ring signature scheme, where for $i \in [1, n]$, $pk_i = g^{x_i} h^{y_i}$ and $sk_i = (x_i, y_i)$ is the private key owned by pk_i. For each of the public keys, $coin_i$ is the total amount of coins associated with the public key.

We also use the linkable ring signature scheme in Sect. 3.2 as the example. Without loss of generality, assume that the user to be attacked holds (pk_1, sk_1), and $coin_1$ is the total amount of coins associated with the public key, where $pk_1 = g^{x_1} h^{y_1}$, and $sk_1 = (x_1, y_1)$. When the user decided to spend his/her coins, he/her collects a list of public keys $Y = \{pk_i\}_{i=1}^{n}$ from the blockchain, where for $i \in [2, n]$, $coin_1 = coin_i$. After all information of the transaction M_1 is prepared, the user runs **Sign**($event$, n, Y_1, sk_1, M_1) to generate a signature $\sigma_1 = (t_1, \overline{x}_1, \overline{y}_1, c_1^{(1)}, \ldots, c_n^{(1)})$ for his/her transaction.

To break the anonymity (Untraceability) of the target user, the remaining members of Y generate transactions and sign their transaction according to the denying attack. Concretely, for $i \in [2, n]$, the user who holds (pk_i, sk_i) will run **Sign**($event$, 1, pk_i, sk_i, M_i) to generate a linkable ring signature $\sigma_i = (t_i, \overline{x}_i, \overline{y}_i, c_1^{(i)}, \ldots, c_n^{(i)})$ with only one ring member. If **Verify** ($event$, 1, pk_i, M_i, σ_i) = **accept**, everyone would be convinced that σ_i was signed by pk_i and pk_i is not the one who signed σ_1. Thus, the anonymity of the target user will be broken. The trick for the attackers to reduce their costs is in their transaction M_i. In a typical ring-signature-based cryptocurrency, *e.g.*, CryptoNote, the public key for receiving coins is generated by the sender rather than the receiver, so that for any two outgoing transactions, nobody could prove that they were sent to the same user (Linkability). Such a technique is called stealth address. Consequently, for $i \in [2, n]$, the user could write in the transaction M_i that, the coins $coin_i$ in pk_i will be transferred to the stealth addresses (public keys) pk_{i1}, …, pk_{im}, where each of them will receive equivalent coins. If only one of the public keys is belonging to the sender, say pk_{i1}, then it is hard for the other users of the blockchain to link pk_1 with pk_{i1}. Thus, when the user decides to spend the coins in pk_{i1}, pk_{i1} is still unlinked. In a much special case, if the coins of the target user $coin_1$ is not a large amount, the attacker who hold pk_i could make none of pk_{i1}, …, pk_{im} to be his public key (*i.e.*, the attacker throws away his/her coins to attack the target user).

The above concrete strategy for launching denying attack seems useful to reduce the costs of an attacker in blockchain system. Moreover, if the blockchain system adopts RingCT or Confidential Transactions [17], which are responsible for hiding coin amounts of transactions, the performance of the above strategy is much better, since attack has the potential to implement the denying attack with the total amount of coins much smaller than $coin_1$.

Against the attack. After all the above discussions, it is easy to see that step 2c in Sect. 4 is a crucial point in the denying attack. The step is to state that an attacker truly holds a public key in the target list (*i.e.*, ring Y). Thus, even though linkable ring signature cannot provide strong anonymity in blockchain applications, one possible approach to prevent denying attack is to forbid a linkable ring signature with only one member (or duplicated members). For example, the miner program in a cryptocurrency system will not package a transaction with such a signature and upload them to blockchain. With this approach, the necessary conditions for the denying attack are collapsed.

6 Conclusion

In this work, we discussed the relation between strong linkability and strong anonymity in a linkable ring signature scheme. We have shown that, even though the scheme in [10] was proven to be strong anonymity under the slightly weaker definition, its anonymity is not as strong as a conventional ring signature could reach to. To address this observation, we proposed an attack to break the strong anonymity of arbitrary linkable ring signature schemes with strong linkability, including the one in [10]. Moreover, such an attack is easily implemented in blockchain applications.

The skills in [10] are elegant, but is perhaps of limited practical use. Their main idea is to employ a Pedersen commitment as a public key of a user, so that even a computationally unbounded adversary is infeasible to

extract the actual private key owned by a signer. Such a skill indeed helps to build a linkable ring signature with strong linkability and anonymity in some weak sense, but the crucial point is that a computationally unbounded adversary does not imply the condition that members will reveal their identities. Moreover, in a strongly anonymous ring signature scheme, anonymity holds even when identities were exposed. Consequently, there is an obvious gap between linkable and conventional ring signatures in anonymity.

To address our idea in detail, we implemented concrete attacks in specific scenarios, such as e-voting and cryptocurrency system, and also discussed the costs and motivations for implementing them. Finally, we suggested a possible approach to prevent the denying attack by break its necessary conditions.

Acknowledgements

This work is supported by Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), the National Natural Science Foundation of China (61972429), the Foundation of Education Department of Hunan Province (21C0208), and the Natural Science Foundation of Hunan Province (2021JJ30741).

References

[1] M. Abe, M. Ohkubo, K. Suzuki, 1-out-of-n Signatures from a Variety of Keys, *Advances in Cryptology - ASIACRYPT 2002*, Queenstown, New Zealand, 2002, pp. 415-432.

[2] M. Bellare, D. Micciancio, B. Warinschi, Foundations of Group Signatures: Formal definitions, Simplified Requirements, and a Construction based on General Assumptions, *Advances in Cryptology - EUROCRYPT 2003*, Warsaw, Poland, 2003, pp. 614-629.

[3] S. Canard, A. Georgescu, G. Kaim, A. Roux-Langlois, J. Traoré, Constant-size Lattice-based Group Signature with Forward Security in the Standard Model, *Provable and Practical Security - ProvSec 2020*, Singapore, 2020, pp. 24-44.

[4] L. Chen, T. P. Pedersen, New Group Signature Schemes, *Advances in Cryptology - EUROCRYPT'94*, Perugia, Italy, 1994, pp. 171-181.

[5] M. F. Esgin, R. Steinfeld, J. K. Liu, D. Liu, Lattice-based Zero-knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications, *Advances in Cryptology - CRYPTO 2019*, CA, USA, 2019, pp. 115-146.

[6] M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, D. Liu, Short Lattice-based One-out-of-many Proofs and Applications to Ring Signatures, *Applied Cryptography and Network Security - ACNS 2019*, Bogota, Colombia, 2019, pp. 67-88.

[7] J. Groth, M. Kohlweiss, One-out-of-many Proofs: or How to Leak a Secret and Spend a Coin, *Advances in Cryptology - EUROCRYPT 2015*, Sofia, Bulgaria, 2015, pp. 253-280.

[8] I. R. Jeong, J. O. Kwon, D. H. Lee, Ring Signature with Weak Linkability and Its Applications, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20, No. 8, pp. 1145-1148, August, 2008.

[9] B. Libert, S. Ling, K. Nguyen, H. Wang, Zero-knowledge Arguments for Lattice-based Accumulators: Logarithmic-size Ring Signatures and Group Signatures without Trapdoors, *Advances in Cryptology - EUROCRYPT 2016*, Vienna, Austria, 2016, pp. 1-31.

[10] J. K. Liu, M. H. Au, W. Susilo, J. Zhou, Linkable Ring Signature with Unconditional Anonymity, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 1, pp. 157-165, January, 2014.

[11] J. K. Liu, V. K. Wei, D. S. Wong, Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups, *Information Security and Privacy - ACISP 2004*, Sydney, Australia, 2004, pp. 325-335.

[12] Z. Lu, Z. L. Jiang, Y. Wu, X. Wang, Y. Zhong, A Lattice-based Anonymous Distributed E-cash from Bitcoin, *Provable Security - ProvSec 2019*, Cairns, QLD, Australia, 2019, pp. 275-287.

[13] Z. Liao, X. Pang, J. Zhang, B. Xiong, J. Wang, Blockchain on Security and Forensics Management in Edge Computing for IoT: A Comprehensive Survey, *IEEE Transactions on Network and Service Management*, Vol. 19, No. 2, pp. 1159-1175, June, 2022.

[14] Z. Liao, S. Cheng, J. Zhang, W. Wu, J. Wang, P. K. Sharma, GpDB: A Graph-partition Based Storage Strategy for DAG-Blockchain in Edge-cloud IIoT, *IEEE Transactions on Industrial Informatics*, (Early Access), March, 2022.

[15] S. Ma, Y. Deng, M. Bai, D. He, J. Zhang, X. Xie, A Practical NIZK Argument for Confidential Transactions over Account-model Blockchain, *Provable and Practical Security - ProvSec 2020*, Singapore, 2020, pp. 234-253.

[16] I. Miers, C. Garman, M. Green, A. D. Rubin, Zerocoin: Anonymous Distributed E-cash from Bitcoin, *Symposium on Security and Privacy - SP 2013*, Berkeley, CA, USA, 2013, pp. 397-411.

[17] S. Noether, A. Mackenzie, Ring Confidential Transactions, *Ledger*, Vol. 1, pp. 1-18, December, 2016.

[18] M. Ober, S. Katzenbeisser, K. Hamacher, Structure and Anonymity of the Bitcoin Transaction Graph, *Future Internet*, Vol. 5, No. 2, pp. 237-250, June, 2013.

[19] T. P. Pedersen, Non-interactive and Information Theoretic Secure Verifiable Secret Sharing, *Advances in Cryptology - CRYPTO 1991*, Santa Barbara, CA, USA, 1991, pp. 129-140.

[20] R. L. Rivest, A. Shamir, Y. Tauman, How to Leak a Secret, *Advances in Cryptology - ASIACRYPT 2001*, Gold Coast, Australia, 2001, pp. 552-565.

[21] D. Ron, A. Shamir, Quantitative Analysis of the Full Bitcoin Transaction Graph, *Financial Cryptography and Data Security - FC 2013*, Okinawa, Japan, 2013, pp. 6-24.

[22] N. v. Saberhagen, CryptoNote v2.0, October, 2013. https://cryptonote.org/whitepaper.pdf

[23] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash: Decentralized Anonymous Payments from Bitcoin, *Symposium on*

Security and Privacy - SP 2014, Berkeley, CA, USA, 2014, pp. 459-474.
[24] J. Wang, B. Wei, J. Zhang, X. Yu, P. K. Sharma, An Optimized Transaction Verification Method for Trustworthy Blockchain-enabled IIoT, *Ad Hoc Networks*, Vol. 119, Article No. 102526, August, 2021.
[25] B. Yin, J. Li, X. Wei, EBSF: Node Characteristics based Block Allocation Plans for Efficient Blockchain Storage, *IEEE Transactions on Network and Service Management*, (Early Access), July, 2022.
[26] F. Zhang, K. Kim, Id-based Blind Signature and Ring Signature from Pairings, *Advances in Cryptology - ASIACRYPT 2002*, Queenstown, New Zealand, 2002, pp. 533-547.
[27] H. Zhang, F. Zhang, H. Tian, M. H. Au, Anonymous Post-quantum Cryptocash, *Financial Cryptography and Data Security - FC 2018*, Nieuwpoort, Curaçao, 2018, pp. 461-479.
[28] J. Zhang, S. Zhong, T. Wang, H. C. Chao, J. Wang, Blockchain-Based Systems and Applications: A Survey, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 1-14, January, 2020.

Biographies

Huang Zhang, received his Ph.D. from the School of Data and Computer Science, Sun Yat-sen University in 2019. He is currently a lecturer at the School of Computer Science and Communication Engineering of Changsha University of Science and Technology. His main research interests include lattice-based cryptography, and zero-knowledge proofs.

Fangguo Zhang received his Ph.D. from the School of Communication Engineering, Xidian University in 2001. He is currently a Professor at the School of Computer Science and Engineering of Sun Yat-sen University, China. He is the co-director of Guangdong Province Key Laboratory of Information Security Technology. His research mainly focuses on cryptography and its applications. Specific interests are elliptic curve cryptography, secure obfuscation, blockchain, anonymity and privacy, etc.

Ke Gu received the Ph.D. degree from the School of Information Science and Engineering, Central South University, Changsha, China, in 2012. He joined the School of Computer and Communication Engineering, Changsha University of Science and Technology, in 2013. He is currently an Associate Professor. His research interests include network and information security.