A simpler construction of traceable and linkable ring signature scheme

Wulu Li[1], Lei Chen[1], Xin Lai[1], Xiao Zhang[1], and Jiajun Xin[1]

Shenzhen Onething Technologies Co., Ltd., Shenzhen, China
liwulu@onething.net

Abstract. Traceable and linkable ring signature scheme (TLRS) plays a major role in construction of regulatable privacy-preserving blockchains, as it empowers the regulator with traceability of signers' identities. A recent work by Li *et al.*[14] gives a modular construction of TLRS by usage of classic ring signature, one-time signature and zero-knowledge proofs, and has security against malicious regulators. In this paper, we introduce sTLRS, a simpler modification of TLRS constructed directly from classic ring signature, without additional one-time signature and zero-knowledge proofs for validity of users' public keys, with public key size reduced by 80% and verification time reduced by over 50%. Moreover, we can further modify the sTLRS to achieve anonymity, unforgeability, linkability, nonslanderability and traceability against malicious regulators.

Keywords: Regulatable blockchains · Privacy preserving · Traceable and linkable ring signature · Malicious regulators.

1 Introduction

Privacy-preserving techniques in blockchain theory has been developed in this decade to provide a potential replacement of traditional blockchain-based cryptocurrencies such as Bitcoin[20] and Ethereum[6]. Privacy-preserving cryptocurrencies, represented by Monero[26] and Zerocash[23], have realized fully anonymous and confidential transactions, which can protect identities for both initiators and recipients in transactions, as well as the transaction amount, making them suitable in various scenarios such as salary, donation, bidding, taxation, etc. A series of works have been proposed during these years such as Confidential Transaction[18], Mimblewimble[13], Dash[9], Monero[26] and Zerocash[23], etc. Among them, Monero uses techniques from Cryptonote[26], Ring-CT[21], Bulletproofs[5] as building blocks, it uses linkable ring signature scheme to hide the identity of initiator, uses Diffie-Hellman key exchange scheme to hide the identity of recipient and uses range proof (Borromean, Bulletproofs) to hide the the amount of transaction.

However, privacy-preserving cryptocurrencies are not regulatable, which may cause abuse of privacy and facilitate illegal transactions by malicious users. It is crucial to develop new regulatory mechanism to realize traceability of users' identities and transaction amount. To solve this issue, a recent work by Li *et al.*[14] proposes the first fully regulatable privacy-preserving blockchain against malicious regulators, their construction contains traceable and linkable ring signature scheme (TLRS), traceable range proofs (TBoRP, TBuRP) and traceable scheme of long-term addresses. Their work is a significant approach to overcome the regulatory barriers on privacy-preserving cryptocurrency. As for construction of TLRS, an additional one-time signature and validity proof of user's public key is needed to prevent slanderability attack (slander honest signers) and traceability attack (escape from regulation), which requires extra storage for public keys and more verification time, making TLRS less efficient than Monero's linkable ring signature (MLSAG). So it is necessary to construct new TLRS with simpler key generation algorithm and less verification time, to support future application of cryptocurrency for high TPS (transactions per second).

1.1 Our Contributions

In this paper, we give a simpler construction of TLRS (named by sTLRS) by removing the one-time signature and validity proofs of public keys $\pi(PK)$ from the key generation algorithm for each user, with public key simplified from $PK = (g_1^x h^a, g_2^a, \pi(g_1^x h^a, g_2^a))$ to $PK = g^x$. This improvement successfully reduce the total size of PK from $(4, 1)$ to $(1, 0)$, where the $(\cdot, \cdot)$ refers to number of elements in $(\mathbb{G}, \mathbb{Z}_q^*)$. Moreover, we reduce the verification computation for each TLRS signature from $(7n + 1, 6n + 1)$ to $(3n + 3, 2n + 1)$, where $(\cdot, \cdot)$ refers to times of exponentiation and multiplication respectively (we take AOS ring signature as component for example).

In addition, we can further modify sTLRS to achieve security (anonymity, unforgeability, linkability, nonslanderability and traceability) against malicious regulator, by adding another generator to the system, with the same public key as sTLRS and verification time increased to $(3n + 5, 2n + 3)$, which is a more compact construction than the original TLRS, while maintaining the same security level.

Simpler Traceable and Linkable Ring Signatures Following the direction of [14], under similar regulatory model, we give a simpler construction of traceable and linkable ring signature scheme (sTLRS) by directly making use of classic ring signature as component, without additional one-time signature and zero-knowledge proofs for public keys. Actually, in our construction, classic ring signature has the same functionalities as linkable (and traceable) ring signature, which will be another independent interest. We give a brief introduction of sTLRS in the following:

1. The public parameter is $(\mathbb{G}, q, g, h = g^y)$, where g is the generator of elliptic curve with prime order, which is uniformly generated by system, y is the regulation trapdoor, generated by the regulator.
2. User generates his (PK, SK) by usage of public parameter, the key generation remains the same as Monero.

3. When signing, the signer publishes a tracing key TK, computes the public keys set L_{RPK}, then uses his private key SK for classic ring signature σ_1, the basis element (generator) for classic ring signature is different from TLRS.
4. The verifier checks whether TK appears in previous signatures to determine whether illegal signature (double spending) occurs. Then computes the public keys set L_{RPK}, checks the validity of classic ring signature σ_1, then outputs the verification results.
5. The regulator can trace the identity of signer by using the trapdoor y and TK.

In the construction of sTLRS, under the discrete logarithm assumption, nobody else can steal the secret keys, nor forge sTLRS signatures of users. In the following we give a brief comparison between TLRS and sTLRS:

1. sTLRS is more efficient than TLRS, no additional one-time signature or zero-knowledge proof is needed, the public key size is reduced by about 80% and verification time is reduced by over 50%.
2. sTLRS can also be modified to be secure against malicious regulator, which means adversary cannot double spend, slander honest users or escape from regulation, with same security property as TLRS.
3. sTLRS can be easily adapted to Monero system, as they share the same key generation algorithm for UTXO public keys. Meanwhile, users' public keys need not to change when regulators change.

A concurrent work[15] gives another construction method of traceable and linkable ring signature, to achieve the traceable Monero system by making use of paring-based accumulators and signature of knowledge. Compared to [15], sTLRS has three main advantages:

1. Construction of sTLRS is modular, we can use arbitrary elliptic-based classic ring signature as component (such as AOS, Ring-CT 3.0, etc.) to achieve smaller signature sizes by choosing the most efficient schemes according to different applications and parameters.
2. Anonymity of sTLRS-based transactions is stronger than [15] for multiple inputs.
3. Security of sTLRS relies on standard assumptions (paring-free).

1.2 Related Works

Ring Signatures Ring signature is a special type of signature scheme, in which signer can sign on behalf of a group chosen by himself, while retaining anonymous within the group. In ring signature, signer selects a list of public key $L_{PK} = \{PK_1, \cdots, PK_n\}$ as the ring elements, and uses his secret key SK_π to sign, verifier cannot determine signer's identity. Ring signature was first proposed by Rivest, Shamir and Tauman[22] in 2001, they constructed ring signature schemes based on RSA trapdoor permutation and Robin trapdoor function, in the random oracle model. In 2002, Abe *et al.*[1] proposed AOS ring signature, which simultaneously supported discrete logarithm (via Sigma protocol) and RSA trapdoor functions (via hash and sign), also in the random oracle model. In 2006, Bender *et al.*[4] introduced the first ring signature scheme in the standard model, by making use of pairing technique. In 2015, Maxwell *et al.*[19] gave Borromean signature scheme, which is a multi-ring signature based on AOS, reduce the signature size from $N + n$ to $N + 1$. It's worth emphasizing that the signature sizes in these schemes are linear to the number of ring elements.

In 2004, building from RSA accumulator, Dodis *et al.*[8] proposed a ring signature scheme with constant signature size in the random oracle model. In 2007, Chandran *et al.*[7] gave a standard model ring signature scheme with $O(\sqrt{n})$ signature size, from pairing technique and require CRS. In 2015, under the discrete logarithm assumption, Groth *et al.*[12] introduced a ring signature scheme with $O(\log n)$ signature size, in the random oracle model. In reality, the schemes mentioned above have shorter signature sizes than Borromean scheme asymptotically when n is sufficient large, but when n is small, these schemes are less efficient as Borromean, and are not used in Monero system.

Linkable Ring Signatures Linkable ring signature is a variant of ring signature, in which the identity of the signer in a ring signature remains anonymous, but two ring signatures can be linked if they are signed by the same signer. Linkable ring signatures are suitable in many different practical applications such as privacy-preserving cryptocurrency (Monero), e-Voting, cloud data storage security, etc. In Monero, linkability is used to check whether double spending happens. The first linkable ring signature scheme is proposed by Liu *et al.*[17] in 2004, under discrete logarithm assumption, in the random oracle model. Later, Tsang *et al.*[25] and Au *et al.*[2] proposed accumulator-based linkable ring signatures with constant signature size. In 2013, Yuen *et al.*[27] gave a standard model linkable ring signature scheme with $O(\sqrt{n})$ signature size, from pairing technique. In 2014, Liu *et al.*[16] gave a linkable ring signature with unconditional anonymity, he also gave the formalized security model of linkable ring signature, which we will follow in this paper. In 2015, Back *et al.*[3] proposed a efficient linkable ring signature scheme LSAG, which shorten the signature size of [17]. In 2016, based on work of Fujisaki *et al.*[10], Noether *et al.*[21] gave a linkable multi-ring signature scheme MLSAG, which support transactions with multiple inputs, and was used by Monero. In 2017, Sun *et al.*[24] proposed Ring-CT 2.0, which is an accumulator-based linkable ring signature with asymptotic smaller signature size than Ring CT, but is less competitive when n is small, besides, the anonymity of Ring-CT 2.0 is lower than Ring-CT for multiple inputs. In 2019, Yuen *et al.*[28] proposed Ring-CT 3.0, a modified Bulletproof-based 1-out-of-n proof protocol with logarithmic size, which has functionality of (linkable) ring signature and is being tested by the Monero group. In 2019, Goodell *et al.*[11] proposed CLSAG, which improved the efficiency of MLSAG.

Traceable and Linkable Ring Signatures Traceable and linkable ring signature is another variant of linkable ring signature, the identity of the signer in a ring signature can be traced by regulator, when a signer signs two ring signatures with one secret key (illegal ring signatures), the signatures will also be linked. In 2019, Li *et al.*[15] gives a construction of traceable Monero to achieve anonymity and traceability of identities by usage of paring-based accumulators, signature of knowledge and verifiable encryption from Ring-CT 2.0, their construction provide the functionality of traceable and linkable ring signature, but relies on extra steps of verifiable encryption and decryption. Besides, in [15], traceability of long-term address depends on zk-SNARKs with CRS, which is inefficient for computation and storage, meanwhile, their work does not provide traceability of transaction amount. In 2019, TLRS[14] is proposed by Li *et al.* in the construction of the first fully regulatable privacy-preserving blockchains against malicious regulators.

In this paper, we introduce sTLRS, which is a modification of TLRS to achieve better efficiency, while under standard assumptions.

1.3 Paper Organization

In section 2 we give some preliminaries; in section 3 we give the construction of the simpler traceable and linkable signature (sTLRS); in section 4 we give the security proof of sTLRS; in section 5 we introduce the modification of sTLRS to achieve security against malicious regulators; in section 6 we give the conclusion.

2 Preliminaries

2.1 Notations

In this paper, we use multiplicative cyclic group $\mathbb{G}$ to represent elliptic group with prime order $|\mathbb{G}| = q$, g is the generator of $\mathbb{G}$, group multiplication is $g_1 \cdot g_2$ and exponentiation is g^a. We use $H(\cdot)$ to represent hash function and *negl* to represent negligible functions. For verifiers, 1 is for *accept* and 0 is for *reject*.

2.2 Classic Ring Signatures

Classic ring signature scheme usually consists of four algorithms: Setup, KeyGen, Rsign, and Verify:

- $\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$ is a probabilistic polynomial time (PPT) algorithm which, on input a security parameter λ, outputs the set of security parameters par which includes λ.
- $(PK_i, SK_i) \leftarrow \mathsf{KeyGen}(\mathsf{Par})$ is PPT algorithm which, on input security parameters par, outputs a private/public key pair (PK_i, SK_i).
- $\sigma \leftarrow \mathsf{Rsign}(SK_\pi, \mu, L_{PK})$ is a ring signature algorithm which, on input user's secret key SK_π, a list of users' public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, where $PK_\pi \in L_{PK}$, and message μ, outputs a ring signature σ.

- $1/0 \leftarrow \mathsf{Verify}(\mu, \sigma, L_{PK})$ is a verify algorithm which, on input message μ, a list of users' public keys L_{PK} and ring signature σ, outputs 1 or 0.

The security definition of ring signature contains *unforgeability* and *anonymity*. Before giving their definitions, we consider the following oracles which together model the ability of the adversaries in breaking the security of the schemes, in fact, the adversaries are allowed to query the four oracles below:

- $c \leftarrow \mathcal{RO}(a)$. *Random oracle*, on input a, random oracle returns a random value.
- $PK_i \leftarrow \mathcal{JO}(\perp)$. *Joining oracle*, on request, adds a new user to the system. It returns the public key PK_i of the new user.
- $SK_i \leftarrow \mathcal{CO}(PK_i)$. *Corruption oracle*, on input a public key PK_i that is a query output of $\mathcal{JO}$, returns the corresponding private key SK_i.
- $\sigma \leftarrow \mathcal{SO}(PK_\pi, \mu, L_{PK})$. *Signing oracle*, on input a list of users' public keys L_{PK}, the public key of the signer PK_π, and a message μ, returns a valid ring signature σ.

Definition 1 (Unforgeability) *Unforgeability for ring signature schemes is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs Setup to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$. $\mathcal{A}$ wins the game if he successfully forges a ring signature $(\sigma^*, L^*_{PK}, \mu^*)$ satisfying the following:*

1. *$\mathsf{Verify}(\sigma^*, L^*_{PK}, \mu^*) = 1$.*
2. *Every $PK_i \in L^*_{PK}$ is returned by $\mathcal{A}$ to $\mathcal{JO}$.*
3. *No $PK_i \in L^*_{PK}$ is queried by $\mathcal{A}$ to $\mathcal{CO}$.*
4. *(μ^*, L^*_{PK}) is not queried by $\mathcal{A}$ to $\mathcal{SO}$.*

We give the advantage of $\mathcal{A}$ in forge attack as follows:

$$\mathrm{Adv}_{\mathcal{A}}^{forge} = \Pr[\mathcal{A} \text{ wins}].$$

A ring signature scheme is unforgeable if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}^{forge} = negl$.

Definition 2 (Anonymity) *Anonymity for ring signature schemes is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs Setup to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$ and $\mathcal{CO}$. $\mathcal{A}$ gives a set of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, $\mathcal{S}$ randomly picks $\pi \in \{1, \cdots, n\}$ and computes $\sigma = \mathsf{Rsign}(SK_\pi, \mu, L_{PK})$, where SK_π is a corresponding private key of PK_π and send σ to $\mathcal{A}$, then $\mathcal{A}$ output a guess $\pi^* \in \{1, \cdots, n\}$. $\mathcal{A}$ wins the game if he successfully guesses $\pi^* = \pi$.*

We give the advantage of $\mathcal{A}$ in anonymity attack as follows:

$$\mathrm{Adv}_{\mathcal{A}}^{anon} = |\Pr[\pi^* = \pi] - 1/n|.$$

A ring signature scheme is anonymous if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}^{anon} = negl$.

In the construction of sTLRS, we use classic ring signature (unforgeable and anonymous in the random oracle model) as component, we may select AOS scheme (linear size) or Ring-CT 3.0 (logarithmic size) in our construction. The choice of ring signature is not restricted, we can choose the most suited ones (most efficient ones) for different ring sizes in different applications, we omit the detailed description of these ring signatures for brevity.

2.3 Linkable Ring Signatures

Based on classic ring signatures, linkable ring signature has the function of linkability, that is, when two ring signatures are signed by the same signer, they are linked by the algorithm Link. We give the definition of Link below:

- $linked/unlinked \leftarrow \mathsf{Link}((\sigma, \mu, L_{PK}), (\sigma', \mu', L_{PK'}))$: verifier checks the two ring signatures are linked or not, output the result.

The security definition of linkable ring signature contains *unforgeability*, *anonymity*, *linkability* and *nonslanderability*. The *unforgeability* is the same as Definition 1, and the *anonymity* is slightly different from Definition 2 with additional requirements that all public keys in L_{PK} are returned by $\mathcal{A}$ to $\mathcal{JO}$ and all public keys in L_{PK} are not queried by $\mathcal{A}$ to $\mathcal{CO}$. In the rest of this paper, we use this modified definition of *anonymity* in TLRS and its security proof.

We give the definition of *linkability* and *nonslanderability* as follows:

Definition 3 (Linkability) *Linkability for linkable ring signature schemes is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs Setup to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$. $\mathcal{A}$ wins the game if he successfully forges k ring signatures $(\sigma_i, L^i_{PK}, \mu_i), i = 1, \cdots, k$, satisfying the following:*

1. *All σ_is are not returned by $\mathcal{A}$ to $\mathcal{SO}$.*
2. *All L^i_{PK} are returned by $\mathcal{A}$ to $\mathcal{JO}$.*
3. *$\mathsf{Verify}(\sigma_i, L^i_{PK}, \mu_i) = 1, i = 1, \cdots, k$.*
4. *$\mathcal{A}$ queried $\mathcal{CO}$ less than k times.*
5. *$\mathsf{Link}((\sigma_i, L^i_{PK}, \mu_i), (\sigma_j, L^j_{PK}, \mu_j)) = unlinked$ for $i, j \in \{1, \cdots, k\}$ and $i \neq j$.*

We give the advantage of $\mathcal{A}$ in linkability attack as follows:

$$\mathrm{Adv}^{link}_{\mathcal{A}} = \Pr[\mathcal{A} \text{ wins}].$$

A linkable ring signature scheme is linkable if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{link}_{\mathcal{A}} = negl$.

The *nonslanderability* of a linkable ring signature scheme is that $\mathcal{A}$ cannot slander other honest users by generating a signature linked with signatures of honest users. In the following we give the definition:

Definition 4 (Nonslanderability) *Nonslanderability for linkable ring signature schemes is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs* Setup *to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$. $\mathcal{A}$ gives a list of public keys L_{PK}, a message μ and a public key $PK_\pi \in L_{PK}$ to $\mathcal{S}$, $\mathcal{S}$ returns the corresponding signature $\sigma \leftarrow \mathsf{Rsign}(SK_\pi, L_{PK}, \mu)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if he successfully outputs a ring signature $(\sigma^*, L^*_{PK}, \mu^*)$, satisfying the following:*

1. $\mathsf{Verify}(\sigma^*, L^*_{PK}, \mu^*) = 1$.
2. PK_π *is not queried by* $\mathcal{A}$ *to* $\mathcal{CO}$.
3. PK_π *is not queried by* $\mathcal{A}$ *as input to* $\mathcal{SO}$.
4. $\mathsf{Link}((\sigma, L_{PK}, \mu), (\sigma^*, L^*_{PK}, \mu^*)) = linked$.

We give the advantage of $\mathcal{A}$ in slandering attack as follows:

$$\mathrm{Adv}_{\mathcal{A}}^{slander} = \Pr[\mathcal{A} \text{ wins}].$$

A linkable ring signature scheme is nonslanderable if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}^{slander} = negl$.

According to [16], linkability and nonslanderability imply unforgeability:

Lemma 5 ([16]) *If a linkable ring signature scheme is linkable and nonslanderable, then it is unforgeable.*

2.4 Traceable and Linkable Ring Signatures

On the basis of security definitions for linkable ring signature, a PPT adversary $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$, and security of TLRS contains unforgeability, anonymity, linkability, nonslanderability and traceability. Considering the existence of regulator, who can trace the identities of signers, so the anonymity only holds for someone not possesses the trapdoor. Moreover, the unforgeability, linkability, nonslanderability remain the same as linkable ring signature, even for malicious regulator (or adversary who corrupts the regulator), he cannot forge signatures of other users, break the linkability and nonslanderability of TLRS, which means that malicious regulator cannot spend money of other users, double spend or slander other users.

Traceability enables regulator with ability to trace signers' identities, for any PPT adversary $\mathcal{A}$ with possession of trapdoor, he cannot escape from regulation. We give the formal definition of traceability as follows:

Definition 6 (Traceability) *Traceability for traceable and linkable ring signature schemes (TLRS) is defined in the following game between the simulator $\mathcal{S}$ and the adversary $\mathcal{A}$, simulator $\mathcal{S}$ runs* Setup *to provide public parameters for $\mathcal{A}$, $\mathcal{A}$ is given access to oracles $\mathcal{RO}$, $\mathcal{JO}$, $\mathcal{CO}$. $\mathcal{A}$ generates a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, $\mathcal{A}$ wins the game if he successfully generates a TLRS signature (σ, L_{PK}, μ) using $PK_\pi \in L_{PK}$, satisfying the following:*

1. $\mathsf{Verify}(\sigma, L_{PK}, \mu) = 1$.

2. $TK_i \neq TK_j$ *for* $1 \leq i < j \leq n$.
3. $\mathsf{Trace}(\sigma, y) \neq \pi$ *or* $\mathsf{Trace}(\sigma, y) = \perp$.

We give the advantage of $\mathcal{A}$ *in traceability attack as follows:*

$$\mathrm{Adv}_{\mathcal{A}}^{trace} = \Pr[\mathcal{A} \text{ wins}].$$

TLRS scheme is traceable if for any PPT adversary $\mathcal{A}$*,* $\mathrm{Adv}_{\mathcal{A}}^{trace} = negl$*.*

We introduce the construction of TLRS[14] for single ring in the following:

- $\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$: system chooses elliptic curve $\mathbb{G}$ and generators $g_1, g_2 \in \mathbb{G}$ independently, the regulator generates $y \in \mathbb{Z}_q^*$ as the trapdoor, computes $h = g_2^y$, system outputs $(\mathbb{G}, q, g_1, g_2, h)$ as the public parameters, in which the regulator dose not know the relation between g_1 and h.
- $(PK, SK) \leftarrow \mathsf{KeyGen}(\mathsf{Par})$:
  1. According to the public parameters $(\mathbb{G}, q, g_1, g_2, h)$, user Alice samples $x, a \in \mathbb{Z}_q^*$, computes $RPK = g_1^x h^a, TK = g_2^a, OPK = h^a$;
  2. Alice gives the validity proof $\pi(RPK, TK) = \pi_{Swit}(g_1^x h^a, g_1^x (g_2 h)^a)$, that is, she gives the switch proof between $RPK = g_1^x h^a$ and $RPK \cdot TK = g_1^x (g_2 h)^a$ that they share the same exponents $(x = x, a = a)$ with basis (g_1, h) and $(g_1, g_2 h)$;
  3. Alice outputs $PK = (RPK, TK, \pi(RPK, TK))$, and retains $SK = (RSK = x, OSK = a)$.
- $\sigma \leftarrow \mathsf{Sign}(SK_\pi, \mu, L_{PK})$:
  1. For a message μ, Alice chooses another $n-1$ users, together with her own public key, to generate a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, where Alice's $PK = PK_\pi \in L_{PK}$;
  2. Alice outputs $OPK = h^{a_\pi}$, then computes
  $$L_{RPK} = \{RPK_1 \cdot OPK^{-1}, \cdots, RPK_n \cdot OPK^{-1}\}$$
  $$= \{g_1^{x_1} h^{a_1 - a_\pi}, \cdots, g_1^{x_n} h^{a_n - a_\pi}\};$$
  3. Alice runs ring signature $\sigma_1 \leftarrow \mathsf{Rsign}(RSK, \mu, L_{RPK}, OPK)$ using L_{RPK} and $RSK = x_\pi$, outputs σ_1;
  4. Alice runs one-time signature $\sigma_2 \leftarrow \mathsf{Osign}(OSK, \sigma_1, OPK)$ using $OPK = h^{a_\pi}$ and $OSK = a_\pi$ (h as the generator);
  5. Alice outputs $\sigma = (\sigma_1, \sigma_2, \mu, L_{PK}, OPK)$.
- $1/0 \leftarrow \mathsf{Verify}(\sigma_1, \sigma_2, \mu, L_{PK}, OPK)$:
  1. Verifier checks the validity of $\pi(RPK_i, TK_i)$ for every $1, \cdots, n$;
  2. Verifier checks $L_{RPK} \overset{?}{=} \{RPK_1 \cdot OPK^{-1}, \cdots, RPK_n \cdot OPK^{-1}\}$;
  3. Verifier checks the validity of ring signature σ_1 and signature σ_2;
  4. If all passed then outputs 1, otherwise outputs 0.
- *linked*/*unlinked* $\leftarrow \mathsf{Link}(\sigma, \sigma')$: For two TLRS signatures $\sigma = (\sigma_1, \sigma_2, \mu, L_{PK}, OPK)$ and $\sigma' = (\sigma_1', \sigma_2', \mu', L_{PK}', OPK')$, if $OPK = OPK'$ then verifier outputs *linked*, otherwise outputs *unlinked*.

- $\pi^* \leftarrow \mathsf{Trace}(\sigma, y)$: For $\sigma = (\sigma_1, \sigma_2, \mu, L_{PK}, OPK)$, the regulator extracts $TK_1, \cdots, TK_n$ from L_{PK}, computes TK_i^y for $i = 1, \cdots, n$, outputs the smallest π^* such that $OPK = TK_{\pi^*}^y$ as the trace result, otherwise outputs $\perp$.

TLRS achieves anonymity, unforgeability, linkability, nonslanderability and traceability against malicious regulators.

3 Simpler Traceable and Linkable Ring Signature

In this section, we give the construction of simpler traceable and linkable ring signature scheme (sTLRS), we modify the key generation algorithm KeyGen and signature algorithm Sign, remove the additional one-time signature and zero-knowledge proofs to achieve better efficiency compared to TLRS. The sTLRS achieves unforgeability, anonymity, linkability, nonslanderability and traceability. In the scenario of privacy-preserving cryptocurrency, unforgeability works for security of users' accounts, anonymity works for anonymity of signers' identities, linkability and nonslanderability works for prevention of double-spending (actively or passively), traceability works for unconditional regulation of signers' identities.

3.1 Construction

In our construction of sTLRS, we use classic ring signature (AOS, Borromean or Ring-CT 3.0) as the ring signature component. Actually, we assume these schemes are anonymous and unforgeable, which makes sTLRS secure under standard assumptions. We give the introduction of sTLRS in the following (single ring as example):

- $\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$: system chooses elliptic curve $\mathbb{G}$ with prime order q and a generator $g \in \mathbb{G}$, the regulator generates $y \in \mathbb{Z}_q^*$ as the trapdoor, computes $h = g^y$, system outputs $(\mathbb{G}, q, g, h)$ as the public parameters.
- $(PK, SK) \leftarrow \mathsf{KeyGen}(\mathsf{Par})$:
  1. According to the public parameters $(\mathbb{G}, q, g, h)$, user Alice samples $x \in \mathbb{Z}_q^*$ as her secret key, then computes $PK = g^x$;
  2. Alice outputs $PK = g^x$, and retains $SK = x$.
- $\sigma \leftarrow \mathsf{Sign}(SK_\pi, \mu, L_{PK})$:
  1. For a message μ, Alice chooses another $n-1$ users, together with her own public key, to generate a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, where Alice's $PK = PK_\pi \in L_{PK}, \pi \in \{1, \cdots, n\}$;
  2. Alice outputs $TK = h^{x_\pi}$, then computes $e_1 = H(L_{PK}, TK, 1)$ and $e_2 = H(L_{PK}, TK, 2)$;
  3. Alice computes and outputs

$$L_{RPK} = \{PK_1^{e_1} \cdot TK^{e_2}, \cdots, PK_n^{e_1} \cdot TK^{e_2}\}$$
$$= \{g^{e_1 x_1} h^{e_2 x_\pi}, \cdots, g^{e_1 x_n} h^{e_2 x_\pi}\};$$

  4. Alice runs classic ring signature $\sigma_1 \leftarrow \mathsf{Rsign}(SK, \mu, L_{RPK}, TK)$ using L_{RPK} and $SK = x_\pi$, outputs σ_1 ($g^{e_1}h^{e_2}$ as the generator);
  5. Alice outputs $\sigma = (\sigma_1, \mu, L_{PK}, TK)$.
- $1/0 \leftarrow \mathsf{Verify}(\sigma_1, \mu, L_{PK}, TK)$:
  1. Verifier computes $e_1 = H(L_{PK}, TK, 1)$ and $e_2 = H(L_{PK}, TK, 2)$;
  2. Verifier checks $L_{RPK} \overset{?}{=} \{PK_1^{e_1} \cdot TK^{e_2}, \cdots, PK_n^{e_1} \cdot TK^{e_2}\}$;
  3. Verifier checks the validity of ring signature σ_1 ($g^{e_1}h^{e_2}$ as the generator);
  4. If all passed then outputs 1, otherwise outputs 0.
- *linked/unlinked* $\leftarrow \mathsf{Link}(\sigma, \sigma')$: For two valid sTLRS signatures $\sigma = (\sigma_1, \mu, L_{PK}, TK)$ and $\sigma' = (\sigma_1', \mu', L_{PK}', TK')$, if $TK = TK'$ then verifier outputs *linked*, otherwise outputs *unlinked*.
- $\pi^* \leftarrow \mathsf{Trace}(\sigma, y)$: For $\sigma = (\sigma_1, \mu, L_{PK}, TK)$, the regulator extracts $PK_1, \cdots, PK_n$ from L_{PK}, computes PK_i^y for $i = 1, \cdots, n$, outputs the smallest $\pi^* \in \{1, \cdots, n\}$ such that $TK = PK_{\pi^*}^y$ as the trace result, otherwise outputs $\perp$.

3.2 Correctness

Theorem 7 (Correctness of sTLRS) *For an honest user Alice in sTLRS, she can complete the ring signature successfully, and regulator can trace her identity correctly.*

Proof. In sTLRS, for Alice's public key $PK = PK_\pi = g^{x_\pi}$, then Alice will output $TK = h^{x_\pi}$ with $L_{RPK} = \{g^{e_1 x_1} h^{e_2 x_\pi}, \cdots, g^{e_1 x_n} h^{e_2 x_\pi}\}$. Since $g^{e_1 x_\pi} h^{e_2 x_\pi} = (g^{e_1} h^{e_2})^{x_\pi}$, then Alice can use $SK = x_\pi$ to generate the ring signature σ_1 ($g^{e_1}h^{e_2}$ as the generator).

For regulator, he can compute $PK_\pi^y = g^{y x_\pi} = h^{x_\pi} = TK$ and then outputs $\mathsf{Trace}(\sigma, y) = \pi$ correctly. □

3.3 Applications in Blockchain

In the applications of privacy-preserving blockchains, using UTXO model, the $PK = g^x$ can be regarded as the UTXO public key generated in the last transaction, which will be published as the UTXO public key $PK = g^x$ during the last transaction. When making transactions, the UTXO owner runs the sTLRS scheme to hide the identity of the real UTXO, he also outputs $TK = h^x$, which is regarded as the Key-image of the UTXO, and Link is used for detection of double-spending. Trace is used for tracing signers' identities by regulator, which brings the regulatory function to the blockchains.

4 Security proofs

In this section we give the security proofs of sTLRS, including anonymity, unforgeability, linkability, nonslanderability and traceability. The security of sTLRS only holds for adversary who does not possess the trapdoor.

4.1 Proof of Anonymity

Theorem 8 (Anonymity) *sTLRS is anonymous for any PPT adversary $\mathcal{A}$ (without possession of trapdoor).*

Proof. Assume $\mathcal{A}$ is playing the game with $\mathcal{S}$ in Definition 2, $\mathcal{A}$ he generates a message μ and a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, where $PK_i = g^{x_i}$, and all PK_is are returned by $\mathcal{JO}$, and $\mathcal{S}$ knows all $SK_i = x_i$.

We consider the following games between $\mathcal{S}$ and $\mathcal{A}$:

– **Game 0**. $\mathcal{S}$ samples $\pi \in \{1, \cdots, n\}$ uniformly at random, publishes $TK = h^{x_\pi}$, computes $e_1 = H(L_{PK}, TK, 1)$, $e_2 = H(L_{PK}, TK, 2)$ and $L_{RPK} = \{g^{e_1 x_1} h^{e_2 x_\pi}, \cdots, g^{e_1 x_n} h^{e_2 x_\pi}\}$, generates the classic ring signature $\sigma_1 = \mathsf{Rsign}(SK, \mu, L_{RPK}, TK)$, outputs $\sigma = (\sigma_1, \mu, L_{PK}, TK)$ to $\mathcal{A}$. When $\mathcal{A}$ receives σ, he gives a guess $\pi^* \in \{1, \cdots, n\}$.
– **Game 1**. $\mathcal{S}$ uniformly at random, samples $\pi \in \{1, \cdots, n\}, r \in \mathbb{Z}_q^*$, publishes $TK = h^r$, computes $e_1 = H(L_{PK}, TK, 1)$, $e_2 = H(L_{PK}, TK, 2)$ and $L_{RPK} = \{g^{e_1 x_1} h^{e_2 r}, \cdots, g^{e_1 x_n} h^{e_2 r}\}$, generates the classic ring signature $\sigma_1 = \mathsf{Rsign}(\mu, L_{RPK}, TK)$ by programming the random oracle, outputs $\sigma = (\sigma_1, \mu, L_{PK}, TK)$ to $\mathcal{A}$. When $\mathcal{A}$ receives σ, he gives a guess $\pi^* \in \{1, \cdots, n\}$.

In the two games above, Game 0 is the real game between $\mathcal{S}$ and $\mathcal{A}$ in sTLRS, and Game 1 is the simulated game in the random oracle model. In game 1, r is uniformly sampled by $\mathcal{S}$, which is statistical independent from the L_{PK}, then $\mathsf{Pr}_{\mathcal{A}}[\pi^* = \pi] = 1/n$.

Then we only need to prove that game 0 and game 1 are computational indistinguishable. If fact, the differences between the two games are generation of TK and L_{RPK}. According to DH assumption, $(g, h, g^{x_\pi}, h^{x_\pi})$ and (g, h, g^{x_π}, h^r) are computational indistinguishable, then $\mathcal{A}$ cannot distinguish h^{x_π} (in game 0) from h^r (in game 1). Then we know $\mathcal{A}$ cannot distinguish $\{g^{e_1 x_1} h^{e_2 x_\pi}, \cdots, g^{e_1 x_n} h^{e_2 x_\pi}\}$ from $\{g^{e_1 x_1} h^{e_2 r}, \cdots, g^{e_1 x_n} h^{e_2 r}\}$, then we know game 0 and game 1 are computational indistinguishable, which finishes the anonymity proof of sTLRS. □

4.2 Proof of Linkability

Theorem 9 (Linkability) *sTLRS is linkable for any PPT adversary $\mathcal{A}$ (without possession of trapdoor).*

Proof. For a PPT adversary $\mathcal{A}$ without possession of the trapdoor y, when $\mathcal{A}$ finished the link game with $\mathcal{S}$ in Definition 3, we assume that $\mathcal{A}$ wins the link game with nonnegligible advantage δ, that is, $\mathcal{A}$ returned k sTLRS signatures $\sigma_i = (\sigma_1^i, \mu_i, L_{PK}^i, TK_i), i = 1, \cdots, k$ (σ_1^is are the classic ring signatures), satisfying the following requirements:

1. All $\sigma_i, i = 1, \cdots, k$ are not returned by $\mathcal{SO}$.
2. All public keys from $L_{PK}^i, i = 1, \cdots, k$ are returned by $\mathcal{JO}$.

3. $\mathsf{Verify}(\sigma_i, L_{PK}^i, \mu_i) = 1$ for $i = 1, \cdots, k$.
4. $\mathcal{A}$ queried $\mathcal{CO}$ less than k times.
5. $\mathsf{Link}((\sigma_i, L_{PK}^i, \mu_i), (\sigma_j, L_{PK}^j, \mu_j)) = \textit{unlinked}$ for $i \neq j \in \{1, \cdots, k\}$.

We first prove a statement that, for a list of users' public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$ returned by $\mathcal{JO}$ with $PK_i = g^{x_i}$, any PPT adversary $\mathcal{A}$ generates a valid sTLRS signature $\sigma \nleftarrow \mathcal{SO}$ if and only if he quires the $\mathcal{CO}$ at least once, except for negligible probability $\epsilon_0 = negl(n)$.

- $\Rightarrow$. If $\mathcal{A}$ gets $SK = x_i$ from $\mathcal{CO}$, and then $\mathcal{A}$ can run the sTLRS signature scheme to generate a valid signature $\sigma = (\sigma_1, \mu, L_{PK}, TK)$.
- $\Leftarrow$. Assume $\mathcal{A}$ did not query the $\mathcal{CO}$ and $\mathcal{SO}$ for $L_{PK} = \{PK_1, \cdots, PK_n\}$ and finished the sTLRS signature over $L_{PK} = \{PK_1, \cdots, PK_n\}$ with non-negligible probability δ_1. We first prove that $\mathcal{A}$ does not know any of the secret keys in L_{PK}. Actually, under the hardness of discrete logarithm, $\mathcal{A}$ cannot compute x_i from $PK_i = g^{x_i}, i = 1, \cdots, n$, then the probability of $\mathcal{A}$ obtaining any of x_i is $\epsilon_1 = negl(n)$.
  Next, according to the assumption that $\mathcal{A}$ generates a valid signature $\sigma = (\sigma_1, \mu, L_{PK}, TK)$, then he must have finished the classic signature σ_1 (with generator $g^{e_1}h^{e_2}$), where $e_1 = H(L_{PK}, TK, 1)$, $e_2 = H(L_{PK}, TK, 2)$. Without loss of generality, we assume $TK = g^s h^t$ output by $\mathcal{A}$, then we have $L_{RPK} = \{g^{e_1 x_1}(g^s h^t)^{e_2}, \cdots, g^{e_1 x_n}(g^s h^t)^{e_2}\}$. Since the classic ring signature scheme achieves unforgeability, and $\mathcal{A}$ finished the classic ring signature σ_1 with L_{RPK} under generator $g^{e_1}h^{e_2}$, then we get $\mathcal{A}$ knows $RSK = z$ for at least one $i \in \{1, \cdots, n\}$ $s.t.$ $g^{e_1 x_i}(g^s h^t)^{e_2} = (g^{e_1}h^{e_2})^z$, except for negligible probability $\epsilon_2 = negl(n)$. We can also assume that $e_1 = 0$ or $e_2 = 0$ happens with negligible probability $\epsilon_3 = negl(n)$, which means $\mathcal{A}$ gets a solution for $g^{e_1(x_i - z) + e_2 s} = h^{e_2(z-t)}$ with nonnegligible probability $\delta_1 - \epsilon_1 - \epsilon_2 - \epsilon_3$, if $t \neq z$, then this contradicts with the hardness of discrete logarithm problems, so we have $t = z$. Then we have $(x_i - t)e_1 + se_2 = 0$, if $s \neq 0$, then $e_2 = e_1 s^{-1}(t - x_i)$, which means e_2 is pre-computed before $\mathcal{A}$ runs the hash function (random oracle), which happens with negligible probability. Then we get $s = 0, z = t = x_i$, which contradicts to the assumptions above. Then we get that $\mathcal{A}$ generates a valid sTLRS signature $\sigma \nleftarrow \mathcal{SO}$ if and only if he quires the $\mathcal{CO}$ at least once, except for negligible probability.

According to the fourth requirement that the number of times of $\mathcal{A}$ querying $\mathcal{CO}$ is $\leq k - 1$, and $\mathcal{A}$ returned k valid sTLRS signatures $\sigma_i = (\sigma_1^i, \mu_i, L_{PK}^i, TK_i)$, $i = 1, \cdots, k$, then we know there are two sTLRS signatures from the same query of $\mathcal{CO}$, saying $SK = z$ from $PK = g^z$, and $\mathcal{A}$ finished two unlinked valid sTLRS signature, then there is at least one $TK_i = g^s h^t \neq h^z$ from the two sTLRS signatures (otherwise they will be linked). We have $L_{RPK} = \{g^{e_1 x_1}(g^s h^t)^{e_2}, \cdots, g^{e_1 x_n}(g^s h^t)^{e_2}\}$, since $\exists j \in \{1, \cdots, n\}$ $s.t.$ $x_j = z$, and $\mathcal{A}$ signs with PK_j, then we have $g^{e_1 x_j}(g^s h^t)^{e_2} = (g^{e_1}h^{e_2})^t g^{e_1(z-t)+e_2 s}$ with $g^s h^t \neq h^z$, if $e_1(z - t) + e_2 s = 0$, then we have $z = t$ and $s = 0$, otherwise e_1 (or e_2) is pre-computed before $\mathcal{A}$ runs the hash function (random oracle), which happens with negligible probability ϵ_1. Then we get $e_1(z - t) + e_2 s \neq 0$, and this means $\mathcal{A}$ can compute x *s.t.* $(g^{e_1}h^{e_2})^x = (g^{e_1}h^{e_2})^t g^{e_1(z-t)+e_2 s}$, otherwise $\mathcal{A}$ will break the unforgeability of classic ring signature, which happens with negligible probability ϵ_2, however, we know that $(g^{e_1}h^{e_2})^x = (g^{e_1}h^{e_2})^t g^{e_1(z-t)+e_2 s}$ implies a non-trivial relationship between g and h, which happens with nonnegligible probability $\delta - k\epsilon_0 - \epsilon_1 - \epsilon_2$, this contradicts to the hardness assumption of discrete logarithm problem, then we finish the linkability proof of sTLRS. □

4.3 Proof of Nonslanderability

Theorem 10 (Nonslanderability) *sTLRS is nonslanderable for any PPT adversary $\mathcal{A}$ (without possession of trapdoor).*

Proof. For a PPT adversary $\mathcal{A}$ without possession of the trapdoor y, when $\mathcal{A}$ finished the slandering game with $\mathcal{S}$ in Definition 4, $\mathcal{A}$ gave a list of public keys L_{PK}, a message μ and a public key $PK_\pi \in L_{PK}$ to $\mathcal{S}$, $\mathcal{S}$ returns the corresponding signature $\sigma \leftarrow \mathsf{Sign}(SK_\pi, L_{PK}, \mu)$ to $\mathcal{A}$. We assume that $\mathcal{A}$ wins the slandering game with nonnegligible advantage δ, that is, $\mathcal{A}$ successfully outputs a ring signature $\sigma^* = (\sigma_1^*, \mu^*, L_{PK}^*, TK^*)$, satisfying the following:

1. $\mathsf{Verify}(\sigma^*, L_{PK}^*, \mu^*) = 1$.
2. PK_π is not queried by $\mathcal{A}$ to $\mathcal{CO}$.
3. PK_π is not queried by $\mathcal{A}$ as input to $\mathcal{SO}$.
4. $\mathsf{Link}((\sigma, L_{PK}, \mu), (\sigma^*, L_{PK}^*, \mu^*)) = linked$.

From the definition of Link, we know that $TK^* = TK = h^{x_\pi}$, since $PK_\pi = g^{x_\pi}$ was not queried by $\mathcal{A}$ to $\mathcal{CO}$ and $\mathcal{SO}$, then $\mathcal{A}$ does not know $SK = x_\pi$ except for negligible probability $\epsilon_0 = negl(n)$ under the hardness of discrete logarithm problems. Then we know $\mathcal{A}$ successfully produced a classic ring signature σ_1^* with nonnegligible advantage $\delta - \epsilon_0$, according to the unforgeability of classic ring signature, we know that $\mathcal{A}$ knows at least one signing key except for negligible probability ϵ_1, that is, there exists $j \in \{1, \cdots, n\}$, $\mathcal{A}$ knows x *s.t.* $(PK_j^*)^{e_1} TK^{e_2} = (g^{e_1}h^{e_2})^x$ with nonnegligible advantage $\delta - \epsilon_0 - \epsilon_1$, where $e_1 = H(L_{PK}^*, TK, 1)$, $e_2 = H(L_{PK}^*, TK, 2)$. Without loss of generality, we assume $PK_j^* = g^s h^t$ output by $\mathcal{A}$, then we have $(g^s h^t)^{e_1} h^{e_2 x_\pi} = (g^{e_1}h^{e_2})^x = (g^{e_1}h^{e_2})^s h^{e_1 t + e_2(x_\pi - s)}$, using similar arguments in Theorem 9, if $e_1 t + e_2(x_\pi - s) = 0$, then we have $x_\pi = s$ and $t = 0$, otherwise e_1 (or e_2) is pre-computed before $\mathcal{A}$ runs the hash function (random oracle), which happens with negligible probability ϵ_2. Then $e_1 t + e_2(x_\pi - s) \neq 0$ and $\mathcal{A}$ gets a non-trivial relationship between g and h with nonnegligible advantage $\delta - \epsilon_0 - \epsilon_1 - \epsilon_2$, which contradicts to the hardness of discrete logarithm problems, then we finish the nonslanderability proof of sTLRS. □

According to lemma 5, we get the unforgeability of sTLRS:

Corollary 11 (Unforgeability) *sTLRS is unforgeable for any PPT adversary $\mathcal{A}$ without possession of trapdoor.*

4.4 Proof of Traceability

Theorem 12 (Traceability) *sTLRS is traceable for any PPT adversary $\mathcal{A}$ (without possession of trapdoor).*

Proof. For a PPT adversary $\mathcal{A}$ without possession of the trapdoor y, when $\mathcal{A}$ finished the tracing game with $\mathcal{S}$ in Definition 6, $\mathcal{A}$ generates a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, we assume that $\mathcal{A}$ wins the tracing game with nonnegligible advantage δ, that is, $\mathcal{A}$ generates a sTLRS signature $\sigma = (\sigma_1, \mu, L_{PK}, TK)$ using $PK_\pi \in L_{PK}$, satisfying the following:

1. $\mathsf{Verify}(\sigma, L_{PK}, \mu) = 1$.
2. $PK_i \neq PK_j$ for $1 \leq i < j \leq n$.
3. $\mathsf{Trace}(\sigma, y) \neq \pi$ or $\mathsf{Trace}(\sigma, y) = \perp$.

It should be emphasized that the TK_i in Definition 6 (related with TLRS) is actually PK_i in sTLRS, and TK in sTLRS is actually OPK in TLRS. We assume $PK_i = g^{x_i} h^{y_i}, i = 1, \cdots, n$ returned by $\mathcal{A}$ without loss of generality, and assume $TK = g^s h^t$. Then we have:

$$L_{RPK} = \{(g^{x_1} h^{y_1})^{e_1} (g^s h^t)^{e_2}, \cdots, (g^{x_n} h^{y_n})^{e_1} (g^s h^t)^{e_2}\}$$

$$= \{g^{e_1 x_1 + e_2 s} h^{e_1 y_1 + e_2 t}, \cdots, g^{e_1 x_n + e_2 s} h^{e_1 y_n + e_2 t}\}.$$

Where $e_1 = H(L_{PK}, TK, 1)$, $e_2 = H(L_{PK}, TK, 2)$, moreover, we assume $e_i \neq 0$ for $i = 1, 2$, except for negligible probability ϵ_0. According to the condition that $\mathcal{A}$ signed σ_1 with PK_π, then we get $\mathcal{A}$ knows the corresponding $SK_\pi = z$, except for negligible probability ϵ_1, under the unforgeability of ring signature, that is:

$$g^{e_1 x_\pi + e_2 s} h^{e_1 y_\pi + e_2 t} = (g^{e_1} h^{e_2})^{e_1^{-1}(e_1 x_\pi + e_2 s)} h^{e_1 y_\pi + e_2 t - e_1^{-1} e_2 (e_1 x_\pi + e_2 s)}$$

$$= (g^{e_1} h^{e_2})^z.$$

In the rest of the proof, we prove that $x_\pi = t$ and $s = y_\pi = 0$. First, if $e_1 y_\pi + e_2 t - e_1^{-1} e_2 (e_1 x_\pi + e_2 s) \neq 0$, following the similar arguments in Theorem 9, we know that $\mathcal{A}$ gets a non-trivial relationship between g and h, this happens with negligible probability ϵ_2 according to the hardness of discrete logarithm problems. Then we get $e_1 y_\pi + e_2 t - e_1^{-1} e_2 (e_1 x_\pi + e_2 s) = 0$ with nonnegligible probability $\delta - \epsilon_0 - \epsilon_1 - \epsilon_2$. Then we have $e_1^2 y_\pi + e_1 e_2 (t - x_\pi) + e_2^2 s = 0$, and e_2 is exactly the solution for equation $sx^2 + e_1(t - x_\pi)x + e_1^2 y_\pi = 0$ in $\mathbb{Z}_q$ after $e_1 = H(L_{PK}, TK, 1)$ was generated, which has at most two solutions when $s(t - x_\pi) \neq 0$ and q is a prime. This means e_2 is pre-determined (1 out of 2) before $\mathcal{A}$ runs the hash function (random oracle), this also happens with negligible probability ϵ_3, then we get $x_\pi = t$ and $s = y_\pi = 0$, which means the equation degenerated to zero. Then we have $PK_\pi^y = g^{ty} = h^t = TK$, then $\mathsf{Trace}(\sigma, y) = \pi$, which contradicted with the assumptions before then we finish the traceability proof of sTLRS. □

5 Modification

5.1 Construction

In this section, we give a modification for sTLRS to achieve security against malicious regulators, which means even for adversary $\mathcal{A}$ with possession of the trapdoor, he still cannot double spent, slander honest users, forge sTLRS signatures nor escape from regulation. Together with the traceable range proof and traceable scheme for long-term addresses[14], we can finally finish the a new construction of fully regulatable privacy-preserving blockchains, with better efficiency and smaller transaction size. In the following we give the detailed description of the modified sTLRS:

- $\mathsf{Par} \leftarrow \mathsf{Setup}(\lambda)$: system chooses elliptic curve $\mathbb{G}$ with prime order q and a generator $g_1 \in \mathbb{G}$, the regulator generates $y \in \mathbb{Z}_q^*$ as the trapdoor, computes $h = g_1^y$, system computes $g_2 = H_1(g_1, h)$ (use hash to point), system outputs $(\mathbb{G}, q, g_1, g_2, h)$ as the public parameters.
- $(PK, SK) \leftarrow \mathsf{KeyGen}(\mathsf{Par})$:
  1. According to the public parameters $(\mathbb{G}, q, g_1, g_2, h)$, user Alice samples $x \in \mathbb{Z}_q^*$ as her secret key, then computes $PK = g_1^x$;
  2. Alice outputs $PK = g_1^x$, and retains $SK = x$.
- $\sigma \leftarrow \mathsf{Sign}(SK_\pi, \mu, L_{PK})$:
  1. For a message μ, Alice chooses another $n-1$ users, together with her own public key, to generate a list of public keys $L_{PK} = \{PK_1, \cdots, PK_n\}$, where Alice's $PK = PK_\pi \in L_{PK}, \pi \in \{1, \cdots, n\}$;
  2. Alice outputs $I = g_2^{x_\pi}, TK = h^{x_\pi}$, then computes $e_1 = H(L_{PK}, I, TK, 1)$, $e_2 = H(L_{PK}, I, TK, 2)$ and $e_3 = H(L_{PK}, I, TK, 3)$;
  3. Alice computes and outputs
  $$\begin{aligned} L_{RPK} &= \{PK_1^{e_1} \cdot I^{e_2} \cdot TK^{e_3}, \cdots, PK_n^{e_1} \cdot I^{e_2} \cdot TK^{e_3}\} \\ &= \{g_1^{e_1 x_1} g_2^{e_2 x_\pi} h^{e_3 x_\pi}, \cdots, g_1^{e_1 x_n} g_2^{e_2 x_\pi} h^{e_3 x_\pi}\}; \end{aligned}$$
  4. Alice runs classic ring signature $\sigma_1 \leftarrow \mathsf{Rsign}(SK, \mu, L_{RPK}, I, TK)$ using L_{RPK} and $SK = x_\pi$, outputs σ_1 ($g_1^{e_1} g_2^{e_2} h^{e_3}$ as the generator);
  5. Alice outputs $\sigma = (\sigma_1, \mu, L_{PK}, I, TK)$.
- $1/0 \leftarrow \mathsf{Verify}(\sigma_1, \mu, L_{PK}, I, TK)$:
  1. Verifier computes $e_1 = H(L_{PK}, I, TK, 1)$, $e_2 = H(L_{PK}, I, TK, 2)$ and $e_3 = H(L_{PK}, I, TK, 3)$;
  2. Verifier checks $L_{RPK} \overset{?}{=} \{PK_1^{e_1} \cdot I^{e_2} \cdot TK^{e_3}, \cdots, PK_n^{e_1} \cdot I^{e_2} \cdot TK^{e_3}\}$;
  3. Verifier checks the validity of ring signature σ_1 ($g_1^{e_1} g_2^{e_2} h^{e_3}$ as the generator);
  4. If all passed then outputs 1, otherwise outputs 0.
- $\mathit{linked/unlinked} \leftarrow \mathsf{Link}(\sigma, \sigma')$: For two valid sTLRS signatures $\sigma = (\sigma_1, \mu, L_{PK}, I, TK)$ and $\sigma' = (\sigma_1', \mu', L_{PK}', I', TK')$, if $I = I'$ then verifier outputs *linked*, otherwise outputs *unlinked*.
- $\pi^* \leftarrow \mathsf{Trace}(\sigma, y)$: For $\sigma = (\sigma_1, \mu, L_{PK}, I, TK)$, the regulator extracts $PK_1, \cdots,$ PK_n from L_{PK}, computes PK_i^y for $i = 1, \cdots, n$, outputs the smallest $\pi^* \in \{1, \cdots, n\}$ such that $TK = PK_{\pi^*}^y$ as the trace result, otherwise outputs $\perp$.

5.2 Correctness

Theorem 13 (Correctness) *For an honest user Alice in the modified sTLRS, she can complete the ring signature successfully, and regulator can trace her identity correctly.*

Proof. In sTLRS, for Alice's public key $PK = PK_\pi = g^{x_\pi}$, then Alice will output $I = g_2^{x_\pi}$ and $TK = h^{x_\pi}$ with $L_{RPK} = \{g_1^{e_1 x_1} g_2^{e_2 x_\pi} h^{e_3 x_\pi}, \cdots, g_1^{e_1 x_n} g_2^{e_2 x_\pi} h^{e_3 x_\pi}\}$. Since $g_1^{e_1 x_\pi} g_2^{e_2 x_\pi} h^{e_3 x_\pi} = (g_1^{e_1} g_2^{e_2} h^{e_3})^{x_\pi}$, then Alice can use $SK = x_\pi$ to generate the ring signature σ_1 ($g_1^{e_1} g_2^{e_2} h^{e_3}$ as the generator).

For regulator, he can compute $PK_\pi^y = g^{y x_\pi} = h^{x_\pi} = TK$ and then outputs $\mathsf{Trace}(\sigma, y) = \pi$ correctly. □

5.3 Applications in Blockchain

In the applications of privacy-preserving blockchains, using UTXO model, the $PK = g_1^x$ can be regarded as the UTXO public key generated in the last transaction, which will be published as the UTXO public key $PK = g_1^x$ during the last transaction. When making transactions, the UTXO owner runs the sTLRS scheme to hide the identity of the real UTXO, he also outputs $I = g_2^x$, which is regarded as the Key-image of the UTXO, and Link is used for detection of double-spending. He also outputs $TK = h^x$, which is used in Trace for tracing signers' identities by regulator, which brings the regulatory function to the blockchains.

5.4 Security

Following the same direction of Theorem 8,9,10,12, we can prove the anonymity, linkability, nonslanderability, unforgeability, traceability of modified sTLRS, for any PPT adversary $\mathcal{A}$ with possession of the trapdoor, detailed proofs will be given in the full version of this paper.

6 Conclusion

In this paper, we give a new construction of simpler traceable and linkable ring signature scheme (sTLRS) by modifying the key generation algorithm and removing the additional one-time signature and zero-knowledge proofs, which reduces the size of PK, shortens the time for transaction verification, realizes the regulatory function for signers' identities, and can prevent the adversary from double spending, escaping from regulation, slandering users or forging signatures. Moreover, we further modify sTLRS to achieve security against malicious regulators, reaching the same security level as in[14]. Our work is a new approach to construct regulatable privacy-preserving blockchains and cryptocurrencies, and is a potential replacement for Monero-type blockchains.

Future Works In the future, we need to study and improve in the following aspects:

1. Study new method to construct traceable range proof with less verification time and smaller size;
2. Study post-quantum ring signatures and range proofs, such as lattice-based, code-based, multi-variant-based and isogen-based schemes to prepare for the future applications and replacement in the era of quantum computing.

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 415–432. Springer (2002)
2. Au, M.H., Chow, S.S., Susilo, W., Tsang, P.P.: Short linkable ring signatures revisited. In: European Public Key Infrastructure Workshop. pp. 101–115. Springer (2006)
3. Back, A.: Ring signature efficiency. Bitcointalk (accessed 1 May 2015) https://bitcointalk. org/index. php (2015)
4. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Theory of Cryptography Conference. pp. 60–79. Springer (2006)
5. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 315–334. IEEE (2018)
6. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper **3**, 37 (2014)
7. Chandran, N., Groth, J., Sahai, A.: Ring signatures of sub-linear size without random oracles. In: International Colloquium on Automata, Languages, and Programming. pp. 423–434. Springer (2007)
8. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 609–626. Springer (2004)
9. Duffield, E., Diaz, D.: Dash: A privacycentric cryptocurrency. No Publisher (2015)
10. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: International Workshop on Public Key Cryptography. pp. 181–200. Springer (2007)
11. Goodell, B., Noether, S., RandomRun: Compact linkable ring signatures and applications. Cryptology ePrint Archive, Report 2019/654 (2019), https://eprint.iacr.org/2019/654
12. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 253–280. Springer (2015)
13. Jedusor, T.E.: Mimblewimble (2016)
14. Li, W., Chen, L., Lai, X., Zhang, X., Xin, J.: Fully regulatable privacy-preserving blockchains against malicious regulators. Cryptology ePrint Archive, Report 2019/925 (2019), https://eprint.iacr.org/2019/925
15. Li, Y., Yang, G., Susilo, W., Yu, Y., Au, M.H., Liu, D.: Traceable monero: Anonymous cryptocurrency with enhanced accountability. IEEE Transactions on Dependable and Secure Computing (2019)

16. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Transactions on Knowledge and Data Engineering **26**(1), 157–165 (2013)
17. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Australasian Conference on Information Security and Privacy. pp. 325–335. Springer (2004)
18. Maxwell, G.: Confidential transactions. URL: https://people. xiph. org/˜ greg/confidential_values. txt (Accessed 09/05/2016) (2015)
19. Maxwell, G., Poelstra, A.: Borromean ring signatures (2015)
20. Nakamoto, S., et al.: Bitcoin: A peer-to-peer electronic cash system (2008)
21. Noether, S., Mackenzie, A., et al.: Ring confidential transactions. Ledger **1**, 1–18 (2016)
22. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 552–565. Springer (2001)
23. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474. IEEE (2014)
24. 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: European Symposium on Research in Computer Security. pp. 456–474. Springer (2017)
25. Tsang, P.P., Wei, V.K.: Short linkable ring signatures for e-voting, e-cash and attestation. In: International Conference on Information Security Practice and Experience. pp. 48–60. Springer (2005)
26. Van Saberhagen, N.: Cryptonote v 2.0 (2013)
27. Yuen, T.H., Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Efficient linkable and/or threshold ring signature without random oracles. The Computer Journal **56**(4), 407–421 (2013)
28. Yuen, T.H., Sun, S.f., Liu, J.K., Au, M.H., Esgin, M.F., Zhang, Q., Gu, D.: Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security (2019)