Computer Systems Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.039908

ARTICLE

A Conditionally Anonymous Linkable Ring Signature for Blockchain Privacy Protection

Quan Zhou[1,*], Yulong Zheng[1], Minhui Chen[2] and Kaijun Wei[2]

[1]School of Mathematics and Information Science, Guangzhou University, Guangzhou, 510006, China

[2]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, 510006, China

*Corresponding Author: Quan Zhou. Email: zhouqq@gzhu.edu.cn

Received: 24 February 2023 Accepted: 27 April 2023 Published: 09 November 2023

ABSTRACT

In recent years, the issue of preserving the privacy of parties involved in blockchain transactions has garnered significant attention. To ensure privacy protection for both sides of the transaction, many researchers are using ring signature technology instead of the original signature technology. However, in practice, identifying the signer of an illegal blockchain transaction once it has been placed on the chain necessitates a signature technique that offers conditional anonymity. Some illegals can conduct illegal transactions and evade the law using ring signatures, which offer perfect anonymity. This paper firstly constructs a conditionally anonymous linkable ring signature using the Diffie-Hellman key exchange protocol and the Elliptic Curve Discrete Logarithm, which offers a non-interactive process for finding the signer of a ring signature in a specific case. Secondly, this paper's proposed scheme is proven correct and secure under Elliptic Curve Discrete Logarithm Assumptions. Lastly, compared to previous constructions, the scheme presented in this paper provides a non-interactive, efficient, and secure confirmation process. In addition, this paper presents the implementation of the proposed scheme on a personal computer, where the confirmation process takes only 2, 16, and 24 ms for ring sizes of 4, 24 and 48, respectively, and the confirmation process can be combined with a smart contract on the blockchain with a tested millisecond level of running efficiency. In conclusion, the proposed scheme offers a solution to the challenge of identifying the signer of an illegal blockchain transaction, making it an essential contribution to the field.

KEYWORDS

Ring signature; conditionally anonymity; blockchain; privacy protection

1 Introduction

Blockchain is a decentralized distributed database that has proliferated in recent times. In 2008, Nakamoto et al. [1] used the blockchain as a crucial part of the cryptocurrency Bitcoin. It enables two users to trade and transfer money using Bitcoin to each other without the limitations of an external third-party trusted center because it is a public transaction platform for international services. Since then, blockchain technology has flourished through the research and development of different cryptocurrencies [2–5]. To improve the performance and flexibility of blockchain, Ethereum [2] as Blockchain 2.0 is proposed, which allows efficient and flexible execution of transactions by deploying

smart contracts. Smart contracts, the underlying technology of Ethereum, can be compared to computer programs that autonomously perform all contract-related processes with associated outcomes. In addition to the two typical public blockchains mentioned above, permissioned blockchains are also gaining traction. Hyperledger Fabric [6] is a blockchain technology project initiated by the Linux Foundation to develop a cross-industry commercial blockchain platform technology. In contrast to the well-known public blockchain, Hyperledger Fabric technology also includes a member management service mechanism to make identity management, network privacy, secrecy, and censorship capabilities more economically appropriate. Blockchain technology has numerous applications, including mobile crowdsensing [7], data sharing [8], healthcare [9], and IoT [10]. However, the development of blockchain also faces many challenges [11], such as security issues [12].

Ring signature, invented by Rivest et al. [13], provides a function of anonymity for users. A ring signature is created when a signer signs a message with a set of public keys it chooses and its public-private key combination. A ring is a collection of signers and the public key sets they have selected. The verifier can only ascertain that one of the ring members is the source of the signature value; they cannot identify the valid signer. To safeguard the user's privacy, the ring signature uses an actual signer whose identity is concealed inside the ring. Ring signature is frequently employed in e-voting [14], e-transactions, and e-money due to their spontaneity and absolute anonymity. Ring signatures can be classified into three categories: identity-based (IBC) ring signatures [15,16], public key infrastructure (PKI) systems [13,17,18], and certificateless (CLC) systems [19–21], depending on the key generation method. Due to the extensive and rapid growth of e-commerce, a significant number of applications have been created that require ring signatures with a wide range of different properties. Ring signature is divided into linkable ring signature [22], deniable ring signature [23], threshold ring signature [24], verifiable ring signature [25], designated-verifier linkable ring signature [26], and so on based on several property attributes. Blockchain systems currently use linkable ring signatures a lot. In 2004, Liu et al. [22] proposed the first linkable ring signature technique and the notion of a linkable ring signature with spontaneity. Based on the ring signature algorithm, the user generates a link to this signature with their private key, which cannot be forged. The ring signature generated by the same signer for different messages has the same link tag, allowing for the link to the signature generated by the same signer. This effectively prevents double-spending attacks on blockchain transactions.

Monero [5] is one of the most successful implementations of blockchain privacy protection mechanisms that use linkable ring signature technology. Its most remarkable feature is the ability to protect the privacy of both parties involved in a transaction. In Monero, a transaction input contains multiple addresses, but only one of them is the actual input. The remaining addresses are combined for obfuscation. Therefore Monero accomplishes the untraceability of transactions and makes it very difficult to tell whether the output of one transaction is the input of another. It leads to the typical blockchain applications for linkable ring signatures, like RingCT1.0 [27], RingCT2.0 [28], and RingCT3.0 [29]. However, in reality, the user of the transaction must be identified in cases of fraud or illegal transactions, hence a conditionally anonymous linkable ring signature should be created to address such issues; Naor [23] first proposes deniable ring signatures for conditional anonymity, but it is an interactive protocol, and the confirmation process is inefficient. While Zheng et al. [30] have constructed a non-interactive protocol for designing conditional anonymous ring signatures, their scheme still requires the signer to send a message to the verifier honestly. As stated earlier, a secure and non-interactive scheme is needed to address the problem of conditional anonymity, where the signer is not required to honestly send a message to the verifier.

This paper constructs a conditionally anonymous linkable ring signature scheme by the idea of the Diffie–Hellman key exchange protocol and Elliptic Curve Discrete Logarithm Assumptions,

which can provide a non-interactive and secure protocol compared to the previous schemes. The contributions of this paper are as follows:

- This paper proposes an efficient conditionally anonymous linkable ring signature scheme, which can directly find the signer of a transaction and thus achieve conditional anonymity when an illegal or malicious transaction occurs in the blockchain. Compared to previous construction methods, the approach outlined in this paper offers a non-interactive, efficient, and secure confirmation process. Additionally, it is linkable to prevent the double-spending attack in the blockchain. This paper also gives detailed security proof of the proposed scheme.
- This paper provides a complete framework that works with blockchain and uses smart contracts to implement the confirmation procedure. The solution of this paper can effectively run through experimental simulations on a personal computer.

The remainder of this paper is organized as follows. The related work is described in Section 2. Some preliminaries about linkable ring signatures and their security definitions are given in Section 3. The concrete algorithm of the scheme and system framework is given in Section 4. The proof of correctness and security analysis of the scheme is discussed in Section 5. Performance and efficiency analysis of the scheme is described in Section 6. Section 7 is the conclusion of the paper.

2 Related Works

The importance of preserving the security and privacy of blockchain users is growing as blockchain technology develops quickly. In 2013, Saberhagen et al. [5] applied a linkable ring signature to a blockchain-based digital currency protocol. They proposed the CryptoNote protocol for Monero, which effectively protects the identity privacy of both parties to a transaction. However, it cannot achieve conditional anonymity. Naor [23] proposed a deniable ring signature system to address conditional anonymity, which used an interactive protocol to identify the signer of the ring signature. Zheng et al. [30] constructed an efficient conditionally anonymous ring signature in the random oracle model. It simply needed one message to be sent from the signer to the verifier, not multiple interactions. Jiang et al. [31] proposed an anonymous authentication mechanism to achieve conditional anonymity and traceability, but it had a considerable amount of consumption in the signature phase. Zhang et al. [32] designed a novel ring signature scheme with conditional anonymity for permissioned blockchains. Nevertheless, interactive protocols have several security flaws, such as real signers interfering with the interaction maliciously and non-signers not taking part. Although Gao et al. [33] first constructed a non-interactive deniable ring signature scheme, it has unfortunately been proven wrong by Jia et al. [34]. Park et al. [35] proposed a repudiable ring signature approach that can overcome these drawbacks, but it had a large signature length and low efficiency. Although Lin et al. [36] proposed a repudiable ring signature with stronger security and logarithmic-size signature, the confirmation process still needed to be improved. As mentioned above, existing ring signature schemes that achieve conditional anonymity are either interactive protocols or must satisfy the signer to honestly send a message to the verifier. Hence it is significant to build an efficient and secure non-interactive protocol to meet the requirements of conditional anonymity. Based on [23,26,32], this paper constructs a conditionally anonymous linkable ring signature scheme with more efficient and secure.

3 Preliminaries

3.1 Linkable Ring Signature

Definition 1: A linkable ring signature scheme for a message space $\mathcal{M}$ and a public key set

$L = \{P_0, P_1, \ldots, P_{n-1}\}$; it has five algorithms: system initialization, key generation, signature generation, signature verification, and linkability. This paper denotes them as LRS.Setup, LRS.KeyGen, LRS.Sign, LRS.Verify and, LRS.Link, respectively, and the details are shown below:

- **LRS.Setup**$(1^\lambda) \rightarrow$ *params* : After inputting a parameter 1^λ, the LRS.Setup can output parameters *params*.
- **LRS.KeyGen**$(1^\lambda, params) \rightarrow (msk, mvk)$: On inputting the security parameter 1^λ and *params*, the LRS.KeyGen outputs the signing key *msk* and public key *mvk*.
- **LRS.Sign**$(, M, n, L, sk_\pi) \rightarrow (\sigma, Q_\pi)$: On inputting the message M, n members of the public key set L, and private key sk_π of the signer, it can output signature σ and linked tag Q_π.
- **LRS.Verify**$(params, M, L, \sigma, Q_\pi) \rightarrow \{0, 1\}$: On inputting the public key set L, a message M, a signature σ, and a linked tag Q_π, the LRS.Verify outputs 1 or 0, where 1 indicates that the signature is valid.
- **LRS.Link**$(L, M_1, M_2, \sigma_1, Q_1, \sigma_2, Q_2) \rightarrow \{0, 1\}$: On inputting two messages M_1 and M_2, a public key set L, two signatures σ_1 and σ_2, and two linked tags Q_1 and Q_2, if σ_1 and σ_2 are valid ring signatures and have the same link tag $Q_1 = Q_2$, it outputs 1, otherwise outputs 0.

3.2 Security Definition

Definition 2: Unforgeability: The unforgeability of the ring signature is defined by the following game between simulator $\mathcal{S}$ and adversary $\mathcal{A}$:

- $\mathcal{S}$ generates parameters *params* to send to $\mathcal{A}$.
- $\mathcal{A}$ will query three random oracles $\mathcal{JO}$, $\mathcal{CO}$, and $\mathcal{SO}$ adaptively, $\mathcal{S}$ returns some designed values and sent them to $\mathcal{A}$.
- $\mathcal{A}$ will output a message M^*, n members of the public key set L^*, and two forged signatures σ_1^* and σ_2^*.

This paper can be sure that $\mathcal{A}$ wins the above game if four of the following conditions are met:

- Two forged signatures σ_1^* and σ_2^* are valid signatures, and they can pass the verification algorithm: **LRS.Verify**$(params, M, L, \sigma, Q_\pi) \rightarrow 1$.
- All public keys of can be queried by random oracle $\mathcal{JO}$.
- $\mathcal{A}$ cannot corrupt the public key of L^* to get the private key.
- σ_1^* and σ_2^* are not obtained by querying the signature random oracle $\mathcal{SO}$.

Definition 3: Anonymity: The anonymity of the ring signature scheme is defined by a game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ with infinite computational power:

- $\mathcal{S}$ generates parameters *params* to send to $\mathcal{A}$.
- $\mathcal{A}$ will query a random oracle $\mathcal{JO}$ and $\mathcal{S}$ returns some designed values to $\mathcal{A}$.
- $\mathcal{A}$ sends a message M^* and n members of the public key set $L^* = \{P_0^*, P_1^*, \ldots, P_{n-1}^*\}$ to $\mathcal{S}$, where all public keys are obtained by querying $\mathcal{JO}$. $\mathcal{S}$ chooses $\pi \in [0, n-1]$ randomly and generates signature **LRS.Sign**$(M, n, L, sk_\pi) \rightarrow (\sigma_\pi, Q_\pi)$, where sk_π is private key of P_π. $\mathcal{S}$ sends σ_π^* to $\mathcal{A}$.
- $\mathcal{A}$ guesses a value $\pi' \in [0, n-1]$.

If the probability that the value π' guessed by $\mathcal{A}$ satisfies $\pi' = \pi$ is less than or equal to $\frac{1}{n}$, then the ring signature scheme satisfies anonymity.

Definition 4: Elliptic Curve Discrete Logarithm Problem (ECDLP): Given any two points P and Q on an elliptic curve $E(F_p)$, solve for the value x that satisfies the equation $Q = x \cdot P$ is unsolvable in polynomial time.

4 Proposed Scheme Method

4.1 Conditionally Anonymous Linkable Ring Signature Scheme

In this section, this paper designs a conditionally anonymous Linkable ring signature (CA-LRS) scheme to protect the transaction privacy of users. The scheme is divided into six algorithms, the first five of which are similar to the classical linkable ring signature, including Setup, Key Generation (KeyGen), Signature, Verification, and Linking. The final algorithm, Confirmation, can find the signer who generated the ring signature and thus achieve conditional anonymity in exceptional cases, such as illegal transactions. Table 1 explains related symbols, and the concrete steps of the algorithms are as follows:

- **Setup:** On inputting parameter $\mathcal{L}$, it outputs parameters $param = \{E, F_p, K, G, q, H_1\}$, where E is an elliptic curve defined over finite field F_p, K is an additive cyclic group about E, G is the generator of K, and q is the order of K. $H_1 : \{0,1\}^* \to Z_q^*$ is a secure hash function.
- **KeyGen:** On inputting public parameters, it generates $S = \{sk_i = d_i\}(i = 0, 1, \ldots n-1) \in Z_q^*$ as private keys for all signers of ring signature. The public key set is $L = \{P_i = d_i \cdot G\}(i = 0, 1, \ldots n-1)$. In proposed scheme, it defines $sk_\pi = d_\pi(\pi \in \{0, 1, \ldots n-1\})$ as private key of signer. It chooses $sk_D = d_D \in Z_q^*$ and $sk_D \notin S$, computing $P_D = d_D \cdot G$ as confirmation public key.
- **Signature:** After signer inputs his private key, public key set L, and message m, it randomly chooses $a, b_\pi, t_\pi \in Z_q^*$ to compute:

$$\hat{P}_D = d_\pi \cdot P_D. \tag{1}$$

$$Q = a \cdot G. \tag{2}$$

$$R = a \cdot \hat{P}_D. \tag{3}$$

$$W = t_\pi \cdot G + b_\pi \cdot P_D. \tag{4}$$

Table 1: Related symbols explanation

Symbols	Explanation
$\mathcal{L}$	Security parameter
E	An elliptic curve
K	An additive cyclic group about E
q	A large prime number
H_1	A secure hash function $H_1 : \{0,1\}^* \to Z_q^*$
d_i	The private key of member i
P_i	The public key of member i

(Continued)

Table 1 (continued)

Symbols	Explanation
d_π	The private key of the signer of the ring signature
d_D	The private key of confirmation
P_D	The public key of confirmation
L	A public key set of ring signature
$\hat{P}_D$	A linkable tag
m	A message
σ	A signature of message m

Then it computes $c_{\pi+1}$ by:

$$c_{\pi+1} = H_1\left(L, m, P_D, \hat{P}_D, Q, R, W\right). \tag{5}$$

Finally, for $i = \pi+1, \ldots n-1, 0, 1, \ldots \pi-1$, it picks $s_i, b_i, t_i \in Z_q^*$ to compute:

$$Q_i = (s_i + c_i)\cdot G + (t_i + c_i)\cdot P_i. \tag{6}$$

$$R_i = (s_i + c_i)\cdot P_D + (t_i + c_i)\cdot \hat{P}_D. \tag{7}$$

$$W_i = t_i \cdot G + b_i \cdot P_D. \tag{8}$$

$$c_{i+1} = H_1\left(L, m, P_D, \hat{P}_D, Q_i, R_i, W_i\right). \tag{9}$$

It sets $s_\pi = a - c_\pi - (t_\pi + c_\pi)\cdot d_\pi$. The generated signature is $\sigma = \left(c_0, \{s_i\}_{i=0}^{n-1}, \{t_i\}_{i=0}^{n-1}, \{b_i\}_{i=0}^{n-1}, \hat{P}_D\right)$.

- **Verification:** After receiving the signature σ of message m, it computes:

$$\begin{cases} Q_i' = (s_i + c_i)\cdot G + (t_i + c_i)\cdot P_i, \\ R_i' = (s_i + c_i)\cdot P_D + (t_i + c_i)\cdot \hat{P}_D, \\ W_i = t_i \cdot G + b_i \cdot P_D, \\ c_{i+1} = H_1\left(L, m, P_D, \hat{P}_D, Q_i', R_i', W_i'\right). \end{cases} \tag{10}$$

The algorithm outputs 1 ("accept") if and only if:

$$c_0 = c_n = H_1\left(L, m, P_D, \hat{P}_D, Q_{n-1}, R_{n-1}, W_{n-1}\right). \tag{11}$$

Otherwise, it outputs 0 ("reject").

- **Linking:** On inputting the generated two signatures $\left(c_0', \{s_i'\}_{i=0}^{n-1}, \{t_i'\}_{i=0}^{n-1}, \{b_i'\}_{i=0}^{n-1}, \hat{P}_D'\right)$ and $^{\infty}\left(c_0'', \{s_i''\}_{i=0}^{n-1}, \{t_i''\}_{i=0}^{n-1}, \{b_i''\}_{i=0}^{n-1}, \hat{P}_D''\right)$ on two identical rings L, the two signature values are substituted into the verification algorithm, and if both output 1, then verify whether $\hat{P}_D' = \hat{P}_D''$ holds, and if it holds, output 1, otherwise output 0.
- **Confirmation:** Once the illegal transaction occurs, the algorithm finds the signer of the ring signature by inputting the private key sk_D. For any $P_i \in L$ (i from 1 to n), it computes $P_D' = sk_D \cdot P_i$, the algorithm will stop when P_D' equals $\hat{P}_D$ or $i = n$.

4.2 Transaction Signature Framework Based on CA-LRS

Based on the framework of CryptoNote [5], this paper introduces the proposed Conditionally Anonymous linkable ring signature (CA-LRS) scheme into the blockchain technology in this section. It records and stores data utilizing ring signature transactions and guarantees the user's anonymity, and can conditionally find the true signer of the ring signature in case of illegal transactions. Since CryptoNote [5] already contains detailed steps, this paper will simply describe them systematically. As shown in Fig. 1, this paper assumes that Alice wants to transfer her cryptocurrency from her address to Bob. The detailed process of the framework consists of the following five steps:

- **Initialization Phase:** Alice and Bob run the Setup algorithm and KeyGen algorithm of the CA-LRS scheme to generate their respective public-private key pairs (pk_A, sk_A) and (pk_B, sk_B). Bob randomly chooses a string s and uses a hash function to compute $hash(s)$, and then sends it to Alice. Alice encrypts the $hash(s)$ with Bob's public key pk_B to get Y as the hidden address and inputs the key image $X = hash(sk_A)$.
- **Execution Phase:** Without the involvement of other parties, Alice generates n-1 transactions with the same value as the output of her transaction and mixes the transactions with Bob in all of these external output transactions. Alice stores a public key P_D from the blockchain. After the hash computation of this transaction, the signature σ is generated by the Signature algorithm of the CA-LRS scheme. Finally, Alice inputs multiple transaction outputs, Y, X, and σ to generate a new transaction tx.
- **Verification Phase:** By running the linking algorithm, miners can verify that the cryptocurrency for the transaction has been spent on preventing the double-spending attack. The miner node then runs the Verification algorithm of the CA-LRS scheme to verify that the signature σ of the transaction tx is valid. The transaction is legitimate and enclosed in a new block if all verification is successful. Otherwise, the transaction will be invalidated.
- **Consensus Phase:** Miners broadcast information among themselves and agree to add new blocks containing transactions to the blockchain through a consensus mechanism. Additionally, the system pays miners to construct new blocks.
- **Confirmation Phase:** Bob checks the output of multiple transactions and compares them by value Y and then accepts the transactions initiated against him. When there is an illegal transaction already in the blockchain network, the user (Alice) who initiated the transaction needs to be found and this phase will be used. After confirmation by the consensus node, Algorithm 1 can be used to check.

Algorithm 1: Checkin(L, $\hat{P}_D$)

Input: L
Output: int
1 for $i = 0$ to $n - 1$ do
2 if $d_D P_i == \hat{P}_D$ then
3 return i;
4 end
5 end
6 return n;

Signer of the ring signature by the integer value returned. Algorithm 1 enables the retrieval of integer i by inputting a public key set of ring signature L and the linkable tag $\hat{P}_D$. If the return value is i ($i \in [0, n-1]$), the owner of the i^{th} public key of the public key set L is the signer of the ring signature.

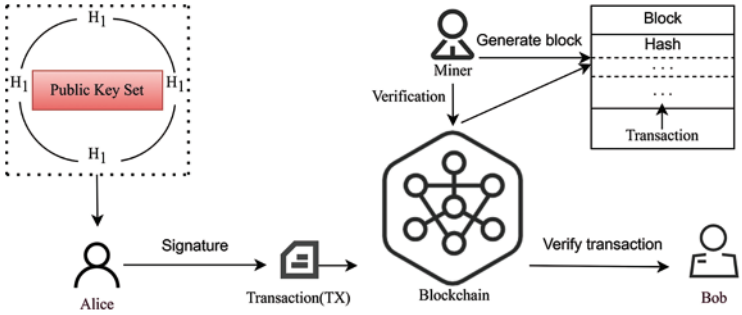

Figure 1: System model of blockchain transactions

5 Security Analysis of Our Scheme

5.1 Correctness

Theorem 1. *Our CA-LRS scheme meets correctness.*

Proof. Assuming that the signature $\sigma = \left(c_0, \{s_i\}_{i=0}^{n-1}, \{t_i\}_{i=0}^{n-1}, \{b_i\}_{i=0}^{n-1}, \hat{P}_D\right)$ is generated by algorithm signature and the verifier uses algorithm verification to verify this signature. When $i = \pi$, there is the following equation:

$$Z'_\pi = (s_\pi + c_\pi) \cdot G + (t_\pi + c_\pi) \cdot P_\pi = [a - (t_\pi + c_\pi) d_\pi] \cdot G + (t_\pi + c_\pi) \cdot d_\pi G = a \cdot G. \tag{12}$$

$$Z''_\pi = (s_\pi + c_\pi) \cdot P_D + (t_\pi + c_\pi) \cdot \hat{P}_D = [a - (t_\pi + c_\pi)\, d_\pi] \cdot P_D + (t_\pi + c_\pi) \cdot d_\pi P_D = a \cdot P_D. \tag{13}$$

$$Z'''_\pi = t_\pi \cdot G + b_\pi \cdot P_D \tag{14}$$

From the above equations, it knows that:

$$c_{\pi+1} = H_1\left(L, m, P_D, \hat{P}_D, Z'_\pi, Z''_\pi, Z'''_\pi\right) = H_1\left(L, m, P_D, \hat{P}_D, Q, R, W\right). \tag{15}$$

Therefore the whole verification process will finally satisfy $c_0 = c_{n-1}$, so it meets correctness.

5.2 Unforgeability

Theorem 2. *If ECDLP is hard, our CA-LRS scheme is unforgeable in the random oracle model.*

Proof. Some definitions of security for proof this paper needs to use from [23,29,37]. Assuming ECDLP is hard to be solved, and $\mathcal{A}$ is a PPT adversary which can forge a valid signature σ of our scheme within a specific time frame, it can construct a simulator $\mathcal{S}$ that uses $\mathcal{A}$ as a subroutine to solve the hard problem. $\mathcal{S}$ first performs the initialization process, generating the required parameters $param = \{E, F_p, \mathbb{G}, G, q\}$, and initial a set $L' = \{P_i\}_{i=0}^{n'-1}$ to interact with $\mathcal{A}$ (The elements in the initial L are n' ECDLP instances). $\mathcal{A}$ makes at most q_H and q_l queries for H_1 and three random oracles $\mathcal{JO}$, $\mathcal{CO}$, and $\mathcal{SO}$. When $\mathcal{A}$ queries H_1 or one of the three random oracles, $\mathcal{S}$ will answer by the following:

- H_1: $\mathcal{S}$ will random choose $g \in Z_q^*$ and output it.
- $\mathcal{JO}$: $\mathcal{S}$ will pick $g \in Z_q^*$, compute $g \cdot G$ and add it to L.
- $\mathcal{CO}$: For $P_j \notin \{P_i\}_{i=0}^{n'-1}$, $\mathcal{S}$ will output the private key d_j of P_j that guarantee $P_j = d_j \cdot G$, otherwise this query wille be terminated.
- $\mathcal{SO}$: When $\mathcal{S}$ receives message m, public key set $L \subset L'$ (The number of elements of L is l), $P_\pi \in L$, and $P_D \in L'$ from $\mathcal{A}$, different signature values are return to $\mathcal{A}$ by the value of P_π. If $P_\pi \notin \{P_i\}_{i=0}^{n'-1}$, $\mathcal{S}$ has d_π that satisfies $d_\pi \cdot G = P_\pi$. So he can use the algorithm signature of the CA-LRS scheme to generate a signature σ and send it to $\mathcal{A}$. Otherwise, he computes $\hat{P}_D = x \cdot P_D$ by chooosing $x \in Z_q^*$ and chooses $\{c_i\}_{i=0}^{l-1}$, $\{t_i\}_{i=0}^{l-1}$, $\{b_i\}_{i=0}^{l-1}$, and $\{s_i\}_{i=0}^{l-1} \in Z_q^*$ randomly. For each $i \in [0, l-1]$, it satisfies the following equation:

$$c_{i+1} = H_1\left(L, m, P_D, \hat{P}_D, (s_i + c_i) \cdot G + (t_i + c_i) \cdot P_i, (s_i + c_i) \cdot P_D + (t_i + c_i) \cdot \hat{P}_D, t_i \cdot G + b_i \cdot P_D\right). \tag{16}$$

Then he sends $\sigma = \left(c_0, \{s_i\}_{i=0}^{l-1}, \{t_i\}_{i=0}^{l-1}, \{b_i\}_{i=0}^{l-1}, \hat{P}_D\right)$ as the signature of the simulation to $\mathcal{A}$. After at most q_l queries for random oracles, it can assume that $\mathcal{A}$ can successfully generate the forged signature. Also the forged signature in general has $L \subseteq \{P_i\}_{i=0}^{n'-1}$, $P_\pi \in L$, and $P_D \in \{P_i\}_{i=0}^{n'-1}$. Since c_{i+1} in a forged signature is a hash value determined by some value, it can use Forking Lemma of [37] to generate two different signatures with

$$\begin{cases} x \cdot G = (s_\pi + c_\pi) \cdot G + (t_\pi + c_\pi) \cdot P_\pi = [s_\pi + c_\pi + d_\pi(t_\pi + c_\pi)] \cdot G, \\ y \cdot P_D = (s_\pi + c_\pi) \cdot P_D + (t_\pi + c_\pi) \cdot \hat{P}_D = [s_\pi + c_\pi + d_\pi\,(t_\pi + c_\pi)] \cdot P_D, \\ z \cdot G = t_\pi \cdot G + b_\pi \cdot P_D = (t_\pi + b_\pi d_D) \cdot G, \end{cases} \tag{17}$$

$$\begin{cases} x \cdot G = (s'_\pi + c'_\pi) \cdot G + (t'_\pi + c'_\pi) \cdot P_\pi = [s'_\pi + c'_\pi + d_\pi(t'_\pi + c'_\pi)] \cdot G, \\ y \cdot P_D = \left(s'_\pi + c'_\pi\right) \cdot P_D + \left(t'_\pi + c'_\pi\right) \cdot \hat{P}_D = \left[s'_\pi + c'_\pi + d_\pi\left(t'_\pi + c'_\pi\right)\right] \cdot P_D, \\ z \cdot G = t'_\pi G + b'_\pi \cdot P_D = (t'_\pi + b'_\pi d_D) \cdot G, \end{cases} \tag{18}$$

If $s_\pi \neq s'_\pi$, the above equation about $x \cdot G$ gives the value of d_π:

$$d_\pi = \frac{s_\pi - s'_\pi + c_\pi - c'_\pi}{t'_\pi - t_\pi + c'_\pi - c_\pi}. \tag{19}$$

If $t_\pi \neq t'_\pi$ and $b_\pi \neq b'_\pi$, the above equation about $z \cdot G$ gives the value of d_D:

$$d_D = \frac{t_\pi - t'_\pi}{b'_\pi - b_\pi}. \tag{20}$$

Therefore, the ECDLP instance is solvable which contradicts the assumption, thus the theorem is proved.

5.3 Anonymity

Unlike general linkable ring signatures, our CA-LRS is conditionally anonymous. The signer of the ring signature can be found when the additional condition of the corresponding private key d_D of P_D is provided; otherwise the generated ring signature is anonymous. The above process of finding a signer is non-interactive, and the proof of conditionally anonymous is given below.

Theorem 3. *Our CA-LRS scheme is conditionally anonymous.*

Proof. Assume an adversary $\mathcal{A}$ who does not know d_D, the following proof shows that the ring signature generated by any member of the ring is statistically indistinguishable from $\mathcal{A}$, and thus the scheme is anonymous. Simulator $\mathcal{S}$ sends parameters $param = \{E, F_p, \mathbb{G}, G, q, H_1\}$ to $\mathcal{A}$ firstly, $\mathcal{A}$ can use up to q_l queries on random oracle $\mathcal{JO}$. $\mathcal{A}$ chooses message m, public key set $L'' = \{P_i\}_{i=0}^{n-1}$ to send to $\mathcal{S}$. $\mathcal{S}$ picks $\omega \in [0, n-1]$ randomly, and computes the ring signature value of message m for ring member $\mathcal{A}_\omega$. When $\mathcal{A}_\omega$ constructs all ring signatures as $(c_i, s_i, t_i, b_i)(i \in [0, n-1], i \neq \omega)$, it is necessary to randomly select $a \in Z_q^*$ and use the appropriate c_ω and t_ω to satisfy the correctness of the ring signature. The correctness of the ring signature is represented by the following equation:

$$s_\omega = a - c_\omega - (t_\omega + c_\omega)d_\omega.$$

For generated n-1 signature values $(c_i, s_i, t_i, b_i)(i \in [0, n-1], i \neq \omega)$, they also need to satisfy that for $i \in [0, n-1]$, the values of $Q_i = (s_i + c_i) \cdot G + (t_i + c_i) \cdot P_i$, $R_i = (s_i + c_i) \cdot P_D + (t_i + c_i) \cdot \hat{P}_D$, and $W_i = t_i \cdot G + b_i \cdot P_D$ guarantee $c_0 = c_n$. Therefore, the probability that $\mathcal{A}_\omega$ constructs n−1 different ring signatures $(c_i, s_i, t_i, b_i)(i \in [0, n-1], i \neq \omega)$ is $Pr\left[\mathcal{A}_\omega\left(\sigma_{dif}\right)\right] = \frac{1}{q-1}\frac{1}{q-2}\cdots\frac{1}{q-n+1}$. The probability that $\mathcal{A}_\omega$ chooses a particular a such that c_ω and t_ω are different from all c_i, $t_i (i \neq \omega)$ is $\frac{1}{q-n}$. This paper can calculate the probability that $\mathcal{A}_\omega$ constructs a valid ring signature is

$$Pr[\mathcal{A}_\omega(\sigma)] = \frac{1}{q-1}\frac{1}{q-2}\cdots\frac{1}{q-n}$$

So the probability of A to distinguish the identity of the real signer of the ring signature is not greater than $\frac{1}{n}$, and our scheme is anonymous. For verifier $\mathcal{V}$ who has private key d_D of P_D, he can compute $P'_D = d_D \cdot P_i$ for any $P_i \in L$, the user of $P_i \in L$ is real signer if and only if $P'_D = \hat{P}_D$.

5.4 Linkability

Theorem 4. *If ECDLP is hard, our CA-LRS scheme is unforgeable in the random oracle model.*

Proof. In the proof of the theorem, this paper continues to follow some of the definitions from the proof of Theorem 2, including a PPT adversary $\mathcal{A}$ that can generate a linkable signature and three random oracles $\mathcal{JO}$, $\mathcal{CO}$, and $\mathcal{SO}$. $\mathcal{A}$ at most queries q_H hash H_1 and q_l random oracles, and they return the same values as the proof in Theorem 2. This paper can construct a simulator $\mathcal{S}$, given n_l ECDLP instances $\{P_i\}_{i=0}^{n_l-1}$, who can output the solution of at least one ECDLP instance by treating $\mathcal{A}$ as a subroutine. After a successful interaction with $\mathcal{S}$, $\mathcal{A}$ will generate some ring signatures denoted as $(\sigma_1, \sigma_2 \ldots \sigma_j)$. It is worth noting that $\mathcal{A}$ at most has corrupted j-1 private keys of ring signatures, and cannot get the private key corresponding to P_D. For the signatures generated by A, there are two different cases. For the first case, $\mathcal{A}$ uses less than j private keys in generated j ring signatures, which results in a pair of signatures generated with the same private key d_π, denoted as $\sigma_{i'}$ and $\sigma_{j'} (1 \leq \sigma_{i'}, \sigma_{j'} \leq j)$. From the proof of Theorem 2 this paper can obtain that:

$$d_\pi = \frac{s_\pi - s'_\pi + c_\pi - c'_\pi}{t'_\pi - t_\pi + c'_\pi - c_\pi} mod\ q \qquad or \qquad d_D = \frac{t_\pi - t'_\pi}{b'_\pi - b_\pi} mod\ q.$$

This means that either the link value of the signature is $\hat{P}_D = d_\pi \cdot P_D$, or the ECDLP instance $p_D = d_D \cdot G$ is solved. Because ECDLP is hard and it cannot solve it, $\sigma_{i'}$ and $\sigma_{j'}$ are linkable is contradictory and the probability that $\mathcal{A}$ generates a signature linked to signature $\sigma_{j'}$ is negligible. For the second

case, all signatures generated by $\mathcal{A}$ are created by different private keys, obviously $\mathcal{A}$ cannot forge a linkable signature in the second case. The above two cases demonstrate that our scheme is linkable.

5.5 *Non-Slanderability*

Theorem 5. *Our scheme is nonslanderable.*

Proof. $\mathcal{S}$ generates parameters $param = \{E, F_p, \mathbb{G}, G, q, H_1\}$ to send to $\mathcal{A}$, and $\mathcal{A}$ will interact with three random oracles $\mathcal{JO}$, $\mathcal{CO}$, and $\mathcal{SO}$ just like in the proof of Theorem 2. $\mathcal{A}$ selects a message m, public key set $L^* = \{P_0, P_1, \ldots, P_{n-1}\}$, and a chosen signer $\pi \in [0, n-1]$ to $\mathcal{S}$, who will generate a corresponding signature $\sigma = \left(c_0, \{s_i\}_{i=0}^{n-1}, \{t_i\}_{i=0}^{n-1}, \{b_i\}_{i=0}^{n-1}, \hat{P}_D\right)$ to return to $\mathcal{A}$. After interacting with three random oracles, $\mathcal{A}$ corrupts P_π^* to get his private key, and finally can forge another signature of the signer π noted as $\sigma = \left(c_0^*, \{s_i^*\}_{i=0}^{n-1}, \{t_i^*\}_{i=0}^{n-1}, \{b_i^*\}_{i=0}^{n-1}, \hat{P}_{D^*}\right)$. $\hat{P}_{D^*} = d_{\pi^*} \cdot P_D$ and $\hat{P}_D = d_\pi \cdot P_D$ can successfully pass the verification of algorithm Linkability of our scheme, and we get $\hat{P}_{D^*} = d_{\pi^*} \cdot P_D = d_\pi \cdot P_D = \hat{P}_D$, and it is oblivious that $d_{\pi^*} P_D = d_\pi P_D$.The final result is $d_{\pi^*} = d_\pi$, which contradicts the fact that $\mathcal{A}$ has not queried d_π. Therefore, our scheme meets non-slanderability.

5.6 *Security of Our Scheme in the Blockchain*

Based on framework Section 4.2, the public-private key pairs of both parties are generated by the Setup algorithm and KeyGen algorithm of our scheme. The Signature algorithm of our scheme generates a signature σ for the final transaction. The security of this part is guaranteed by the security of schemes like [5,7,32]. To finding a real signer of ring signature in the blockchain, it needs to use the consensus algorithm of the blockchain or get the private key corresponding to the P_D, so the adversary cannot query the real signer unless it can cheat the consensus nodes of the blockchain or solve the discrete logarithm problem. For the double-spending attack, the nodes within the blockchain system are capable of detecting double-spending attacks by verifying if the links in the ring signatures of two different transactions are identical. As a result, the system can effectively prevent such attacks.

6 Performance Analysis

In this section, this paper analyzes the computation complexity of our scheme and implement it on a personal computer.

6.1 *Complexity Analysis*

Some of the notation this paper needs to use for complexity analysis are given below:

- T_h:Time costs to run one hash function H_1.
- T_{Gmul}:Time costs of running one multiplication operation in additive group G.
- T_p:Time costs of running one bilinear pair operation.
- T_m: Time costs of running one multiplication operation in field Z_q^*.
- T_e: Time costs of running one exponentiation operation in field Z_q^*.
- n: The size of Public key set.
- $\mathcal{L}$: Security parameter.

This paper provides a theoretical analysis of the primary algorithms used in some schemes, as given in Table 2. Theoretically, the consumption of algorithm Signature and Verification is nearly equal and their values are $6(n-1)T_{Gmul} + nT_h$ and $6nT_{Gmul} + nT_h$, respectively. For algorithm confirmation, this paper takes into account the consumption in the worst case, where all ring signature members

need to be checked. According to Table 2, our method has a faster theoretical efficiency than other schemes on algorithm confirmation, where all schemes consider the worst case. Our approach only costs nT_{Gmul}, which is significantly cheaper than some of the previous constructions. Incidentally, this paper sacrifices the efficiency of the signature and verification process to improve the efficiency of the validation process, which results in a less efficient signature and verification process for our scheme than other schemes, and it requires internal storage of a private key and ensures that the private key is not compromised. Most importantly, our method is non-interactive and does not request for any ring members to participate, which is a security that interactive protocols do not offer when an illegal transaction takes place and needs to query the signer of the ring signature without being spoofed by a malicious signer. Additionally, our scheme is linkable to prevent the double-spending attack in blockchain transactions. The particular advantages of our scheme are shown in Table 3.

Table 2: Theoretical cost analysis of different algorithm

Schemes	Signature	Verification	Confirmation
Ours	$6(n-1)T_{Gmul}+nT_h$	$6nT_{Gmul}+nT_h$	$4n\mathcal{L}T_e$
[23]	$4(n-\mathcal{L})T_e$	$4nT_e$	$n(3T_p+6T_e)$
[30]	$T_p+4T_e+nT_m$	$3T_p+3T_e+nT_m$	$n(6T_p+T_e+3T_m)$
[31]	$3T_p+4T_e+3nT_m$	$2T_p+3T_e+nT_m$	$n(7T_e+3T_m)$
[32]	$5T_e+(n+2)T_m$	$4T_e+(n+1)T_m$	nT_{Gmul}

Table 3: Comparison of the advantages for different schemes

Schemes	Anonymity	Type of confirmation	Linkability
[23]	Yes	Interactive	No
[30]	Yes	Non-interactive	No
[31]	Yes	Interactive	No
[32]	Yes	Interactive	No
Ours	Yes	Non-interactive	Yes

6.2 Implementation

This paper used the VMware Workstation Pro experiment with AMD CPU Ryzen 5 5600H Radeon Graphics @ 3.30 GHz and 16.0 GB RAM. This paper deployed Hyperledger Fabric v1.4.0 on Ubuntu 18.04 to test the smart contract. This paper used Pypbc of Python 3.6.9 to simulate our scheme and compared schemes, running 200 simulations before averaging the results. Notably, this paper uses the BLS12 curve to support bilinear pairings, and the form of the curve is $y^2 = x^3 + 15$ defined over a finite field F_q^*, where q is a prime and $|q| = 383$, and choose SHA-256 as the hash function this paper uses. The time consumption of the signature and verification procedures in our approach, as shown in Figs. 2 and 3, is almost linear with the ring size. When the ring size is 8, the time consumption of the signature and verification algorithms respectively are 116 ms and 170 ms, which is a relatively small time cost. The signature and verification algorithms take 231 ms and 304 ms, respectively, to run when the ring size is 16. When the ring size is 40, the algorithms for signature and verification execute in 560 ms and 791 ms, respectively. The time costs of signature and verification in our method are slightly

slower but still within a suitable time range because this paper slightly increases the computation to increase the security and efficiency of the validation process.

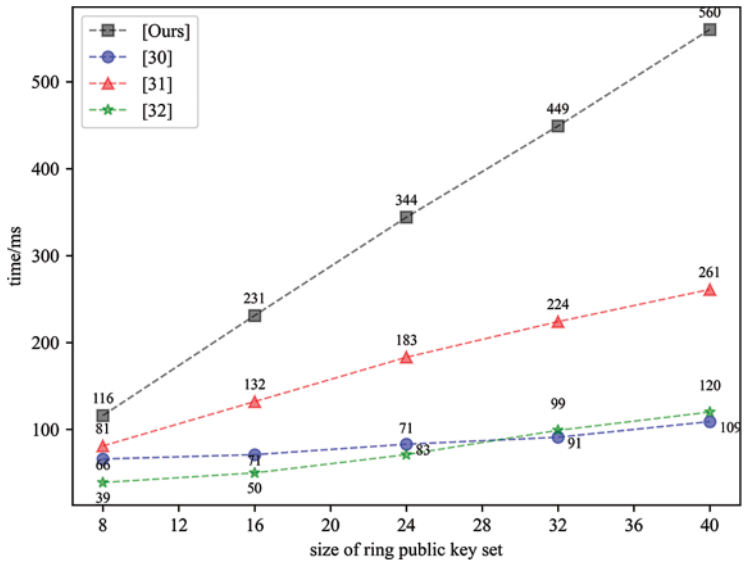

Figure 2: Running times of signature for different ring size

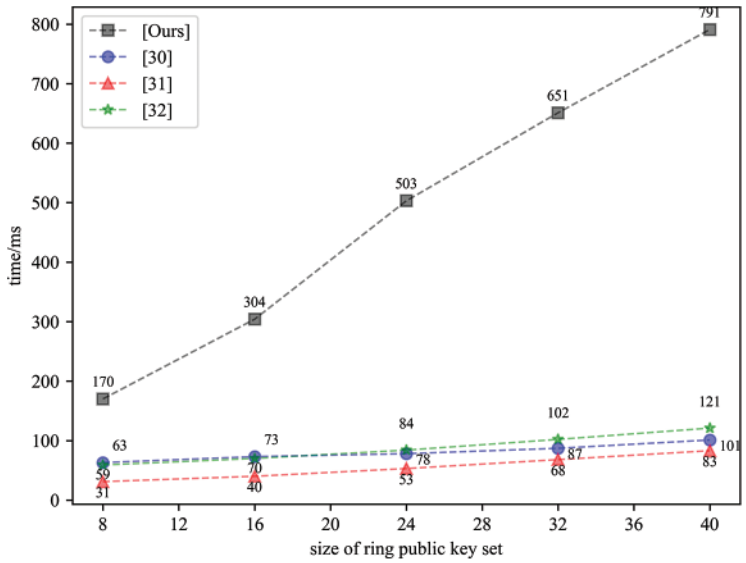

Figure 3: Running times of verification for different ring size

As shown in Fig. 4, all schemes simulations were run on the aforementioned personal computer using the same environment. The Confirmation algorithm of our scheme takes far less time than those of other schemes. When the ring size is 4, the time consumption of [30–32], and our scheme are 36, 43, 25, and 2 ms, respectively. The time consumption of [30–32], and our scheme are 215, 300, 141, and 12 ms when the ring size is 24. Time consumption for [30–32], and our scheme when the ring size is 48 are 428, 512, 290, and 26 ms, respectively. The main reason for the larger time consumption of schemes

[30,31] is the use of bilinear pair. For scheme [32], its confirmation algorithm is an interactive protocol, and when querying the signer of the ring signature in the worst case, it interacts more often and thus spends more time. From the results in Fig. 4, it can see that the confirmation process of proposed scheme is more efficient.

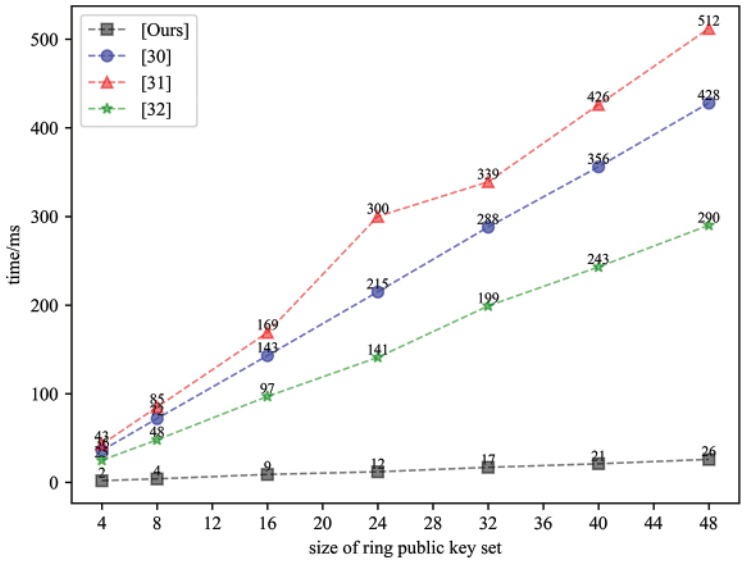

Figure 4: Running times of confirmation for different schemes

This paper created a smart contract in Hyperledger Fabric v1.4.0 and calculated the time costs with various ring sizes. As shown in Fig. 5, the time costs for calling the functions of the smart contract are reasonable.

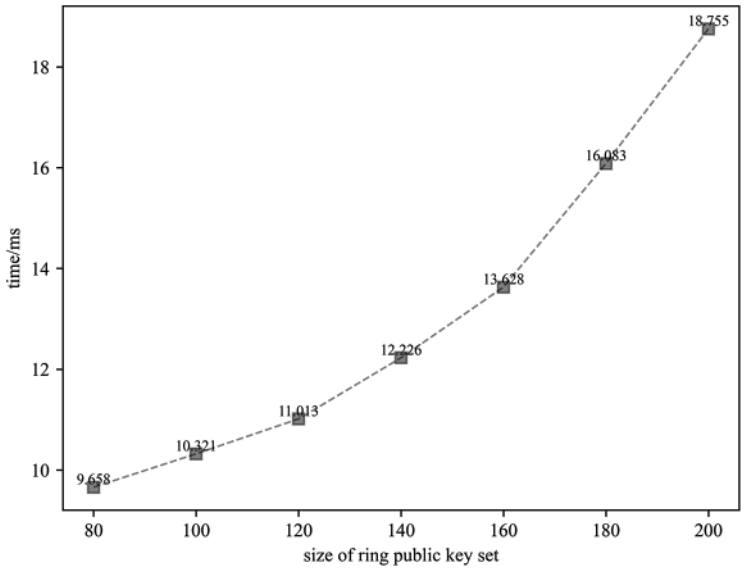

Figure 5: Time costs of invoking smart contract with different ring size

7 Conclusion

With the rapid development of blockchain applications, the issue of blockchain privacy protection and security in real-life situations becomes increasingly essential. Although a blockchain framework using ring signature technology can safeguard the confidentiality of the identities of both parties to a transaction, this paper still needs to identify the user involved in any illegal transactions. This paper proposes a secure conditionally anonymous linkable ring signature scheme to solve such a problem. The scheme provided in this paper offers a non-interactive and secure confirmation mechanism compared to previous constructions. Proposed scheme is proven to be secure under Elliptic Curve Discrete Logarithm Assumptions. Furthermore, this paper analyzed the performance of proposed scheme. The advantage of proposed scheme over other schemes is that the confirmation algorithm is non-interactive and can provide a link to prevent the double-spending attack in the blockchain. The confirmation algorithm consumes only 2, 16, and 24 ms for ring sizes of 4, 24, and 48, respectively. Designing secure and non-interactive protocols to guarantee conditional anonymity while ensuring that signature and verification algorithms are efficient is the focus of future research.

Acknowledgement: This work is supported in part by the National Key R&D Program of China and National Natural Science Foundation of China.

Funding Statement: This research was funded by the National Key R&D Program of China (Grant Number 2021YFA1000600) and National Natural Science Foundation of China (Grant Number 12171114).

Author Contributions: Study conception and design: Z. Quan, Z. Yulong; data collection: C. Minhui; analysis and interpretation of results: W. Kaijun; draft manuscript preparation: Z. Yulong. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This article does not involve data availability and this section is not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Nakamoto and A. Bitcoin, *A Peer-to-Peer Electronic Cash System*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Yellow Paper*, Ethereum Project, Zug, Switzerland, 2014.

[3] E. Ben-Sasson, "Decentralized anonymous payments from bitcoin," in *2014 IEEE Symp. on Security and Privacy*, San Jose, California, pp. 459–474, 2014.

[4] D. Hopwood, S. Bowe, T. Hornby and N. Wilcox, *Zcash Protocol Specification*, 2017. [Online]. Available: https://cryptopapers.info/zcash_protocol/

[5] N. van Saberhagen, in *Cryptonote v2.0*, 2013. [Online]. Available: https://bytecoin.org/old/whitepaper.pdf

[6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis *et al.,* "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. of the Thirteenth EuroSys Conf., 2018*, New York, USA, pp. 1–15, 2018.

[7] W. Z. Wang, Y. Q. Yang, Z. M. Yin, K. Dev, X. K. Zhou *et al.,* "BSIF: Blockchain-based secure, interactive, and fair mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3452–3469, 2022.

[8] L. J. Zhang, Y. F. Zou, M. H. Yousuf, W. Z. Wang, Z. L. Jin *et al.,* "BDSS: Blockchain-based data sharing scheme with fine-grained access control and permission revocation in medical environment," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 5, pp. 1634–1652, 2022.
[9] C. Y. Li, M. X. Dong, J. Li, G. Xu, X. B. Chen *et al.,* "Efficient medical big data management with keyword-searchable encryption in healthchain," *IEEE Systems Journal*, vol. 16, no. 4, pp. 5521–5532, 2022.
[10] W. Z. Wang, H. K. Huang, Z. M. Yin, T. R. Gadekallu, M. Alazab *et al.,* "Smart contract token-based privacy-preserving access control system for industrial internet of things," *Digital Communications and Networks*, vol. 9, no. 2, pp. 337–346, 2022.
[11] Z. B. Zheng, S. A. Xie, H. N. Dai, X. P. Chen and H. M. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
[12] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur and H. N. Lee, "Systematic review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605–6621, 2022.
[13] R. L. Rivest, A. Shamir and Y. Tauman, "How to leak a secret," In: C. Boyd (Ed.), *Advances in Cryptology–ASIACRYPT 2001*, pp. 552–565, Gold Coast, Qld, Australia: Springer, 2001.
[14] Y. F. Wu, "An e-voting system based on blockchain and ring signature," Master Dissertation, University of Birmingham, UK, 2017.
[15] Y. X. Sang, Z. W. Li, L. L. Zhang, H. Jiang and K. C. Li, "Lattice-based identity-based ring signature without trapdoors," *International Journal of Embedded Systems*, vol. 11, no. 3, pp. 386–396, 2019.
[16] M. Nassurdine, H. Zhang and F. G. Zhang, "Identity based linkable ring signature with logarithmic size," in *Information Security and Cryptology: 17th Int. Conf., Inscrypt 2021*, Virtual Event, pp. 42–60, 2021.
[17] E. Fujisaki and K. Suzuki, "Traceable ring signature," *Public Key Cryptography–PKC 2007*, pp. 181–200, 2007.
[18] S. Noether and B. Goodell, "Triptych: Logarithmic-sized linkable ring signatures with applications," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, Guildford, UK, pp. 337–354, 2020.
[19] S. Chow and W. Yap, "Certificatelessring signature," *Cryptology ePrint Archive*, Report 2007/236, 2007.
[20] Y. Y. Zhang, J. W. Zeng, W. Li and H. L. Zhu, "A certificateless ring signature scheme with high efficiency in the random oracle model," *Mathematical Problems in Engineering*, vol. 2017, 2017.
[21] L. Z. Deng, H. Y. Shi and Y. Gao, "Certificateless linkable ring signature scheme," *IEEE Access*, vol. 2020, no. 8, pp. 54641–54651, 2020.
[22] J. K. Liu, V. K. Wei and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," *Australasian Conference on Information Security and Privacy*, vol. 2004, no. 4, pp. 325–335, 2004.
[23] M. Naor, "Deniable ring authentication," in *CRYPTO 2002: 22nd Annual Int. Cryptology Conf. Santa Barbara*, California, USA, pp. 481–498, 2002.
[24] E. Bresson, J. Stern and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *CRYPTO 2002: 22nd Annual Int. Cryptology Conf. Santa Barbara*, California, USA, vol. 2002, pp. 465–480, 2002.
[25] J. Lv and X. Wang, "Verifiable ring signature," in *CANS'03–Third Int. Workshop on Cryptology and Network Security, DMS Proc.*, USA, pp. 663–667, 2003.
[26] P. Behrouz, P. Grontas, V. Konstantakatos, A. Pagourtzis and M. Spyrakou, "Designated-verifier linkable ring signatures," *Information Security and Cryptology*, Seoul, South Korea, vol. 2022, pp. 51–70, 2022.
[27] S. Noether and A. Mackenzie, "Ring confidential transactions," *Ledger*, vol. 2016, no. 1, pp. 1–18, 2016.
[28] S. F. Sun, M. H. Au, J. K. Liu and T. H. Yuen, "RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Computer Security–ESORICS 2017: 22nd European Symp. on Research in Computer Security*, Oslo, Norway, vol. 2017, pp. 456–474, 2017.
[29] T. H. Yuen, S. F. Sun, J. K. Liu, M. H. Au, M. F. Esgin *et al.,* "Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security," in *Int. Conf. on Financial Cryptography and Data Security*, vol. 2020, pp. 464–483, 2020.
[30] S. Zheng, S. Jiang and Z. Qin, "An efficient conditionally anonymous ring signature in the random oracle model," *Theoretical Computer Science*, vol. 2012, no. 461, pp. 106–114, 2012.

[31] Y. Z. Jiang, M. X. He, X. H. Zhang and L. Xiong, "Blockchain-based anonymous authentication mechanism with semi-ttp for vanets," in *2021 IEEE 6th Int. Conf. on Cloud Computing and Big Data Analytics (ICCCBDA)*, Chengdu, China, vol. 2021, pp. 657–666, 2021.
[32] X. Zhang and C. Ye, "A novel privacy protection of permissioned blockchains with conditionally anonymous ring signature," *Cluster Computing*, vol. 25, no. 2, pp. 1221–1235, 2022.
[33] W. Gao, L. Chen, Y. P. Hu, C. J. P. Newton, B. C. Wang *et al.,* "Lattice-based deniable ring signatures," *International Journal of Information Security*, vol. 18, no. 3, pp. 355–370, 2019.
[34] H. W. Jia and C. M. Tang, "Cryptanalysis of a non-interactive deniable ring signature scheme," *International Journal of Information Security*, vol. 20, no. 1, pp. 103–112, 2021.
[35] S. Park and A. Sealfon, "It wasn't me! repudiability and claimability of ring signatures," in *Advances in Cryptology–CRYPTO 2019: 39th Annual Int. Cryptology Conf.*, Santa Barbara, CA, USA, vol. 2019, pp. 159–190, 2019.
[36] H. Lin and M. Q. Wang, "Repudiablering signature: Stronger security and logarithmic-size," *Computer Standards & Interfaces*, vol. 80, 2022.
[37] J. Herranz and G. Sáez, "Forking lemmas for ring signature schemes," in *Int. Conf. on Cryptology in India*, New Delhi, India, vol. 2003, pp. 266–279.