Optimizing Cryptographic Techniques for Enhanced Privacy and Efficiency: A Literature Review on Ring Signature Schemes and Their Applications

[Xiangyu Hui u7238607]

August 29, 2023

1 Introduction

This comprehensive literature review supports our research proposal "Optimized DualRing Signature Scheme: An Integrated Inner Product Argument for Compact and Efficient Anonymous Signature." The proposal seeks to explore and address a crucial gap in the current DualRing signature scheme, aiming to enhance its efficiency and compactness, which are vital attributes for various privacy-preserving applications such as cryptocurrencies and e-voting systems.

The DualRing signature scheme [16], a recent advancement in cryptographic techniques, provides an excellent balance between signature compactness and robust anonymity. However, its construction still leaves room for optimization, specifically in its separate usage of a sum argument zero-knowledge approach, which adds three additional elements to the signature size. This overhead potentially restricts the scheme's efficiency, particularly in large-scale or resource-constrained applications. This research proposal is dedicated to finding a solution for integrating the inner product argument directly into the signature generation process, potentially achieving a more compact and efficient signature size.

Through this literature review, we provide a comprehensive exploration of the advancements in cryptographic techniques, particularly in ring signatures and their applications in modern digital platforms such as Confidential Transactions (CT) in Bitcoin and electronic voting systems. Furthermore, we delve into the current gaps and future direction of research within the domain of ring signatures. Our study presents an examination of the historical development and the state-of-the-art ring signature schemes, their strengths and weaknesses, and their practical implementations.

As a result, this review serves as a strong foundational support for our proposal, illustrating the relevance of our research objective in the context of existing literature, current technological demands, and the challenges that need to be addressed. Through the proposed research, we aspire to contribute significantly to the domain of cryptographic techniques, fortifying the balance between anonymity, efficiency, and security in various digital platforms.

2 Definition

A ring signature is a cryptographic scheme that allows a member of a group of users to sign a message anonymously, while ensuring that the signature can be verified by any observer. In this scheme, each user's public key is openly accessible, while their secret key remains private. The ring signature mechanism consists of the following functions:

SETUP(λ): The SETUP function generates all security parameters based on the input λ, which determines the size of the keys and the level of cryptographic security. This function also generates public-secret key pairs for all users in the given group.

SIGN(m, PKs, sk): A user with a valid secret key can use the Sign function to create a ring signature for a message m. This function takes as input the message m, the set of public keys PKs for the group, and the secret key sk owned by the signing user. The function returns a ring signature, denoted as σ.

VERIFY(m, PKs, σ): The Verify function allows any verifier to check if a message has been signed by a valid user within the group. The function takes as input the message m, the set of public keys PKs, and the ring signature σ. If the signature was created by a valid user, the Verify function returns true; otherwise, it returns false.
These functions form the basis of the ring signature scheme, enabling anonymous signing and verification of messages while preserving the confidentiality of the individual signers.

A robust Ring Signature scheme should meet the following security properties based on[2]:

- Completeness: For any signature σ and corresponding message, if the signature is valid, then the probability that the verifier accepts the proof provided by the signer is (close to) 1, assuming that both the signer and verifier are following the scheme correctly. Some models have more stringent requirements for completeness, requiring perfect completeness, which means that given a valid signature, the probability of the verifier accepting the proof is 1.

- Anonymity: The probability for a verifier to detect the identity of the signer during verification should be less than 1/n, where n is the number of users in the set. When the verifier is included in the set, the probability to detect the identity should be less than 1/(n-1). In some strict models, the probability for any adversarial verifier to detect the identity of the signer should be negligibly small.

- Unforgeability: For any adversary unqualified signer, the probability for a verifier to accept the adversary's generated signature is negligibly small. This means that an adversary cannot forge a valid signature.

3 Ring Signature Algorithms

3.1 Shamir's Ring Signature

Rivest, Shamir, and Tauman first proposed the concept of ring signatures[14], which is based on the RSA encryption (Rivest-Shamir-Adleman) cryptosystem. This widely-used public key

cryptosystem relies on the product of two large prime numbers, n, and a common exponent, e, for encryption. Decryption is performed using n and the modular multiplicative inverse of e. The scheme's security is derived from the mathematical difficulty of factoring large integers. For more in-depth information on RSA, readers are referred to [13]. The original ring signature scheme's sign and verify functions are as follows:
SIGN(m, PKs, sk):

- Hash the message m to obtain the key, i.e., $k = hash(m)$;
- Randomly choose x_j for all users in the set, except the signer i, and compute$y_j = g_j(x_j)$, where g_i is a trapdoor function utilizing RSA encryption.
- Randomly choose a value v and ensure the following equation holds: $v = C_{k,v}(y_1, y_2...y_i...y_n) = E_k(y_n \oplus E_k(y_{n-1} \oplus \dots E_k(y_1)\dots))$, where E_k is symmetric encryption based on key k.
- Solve for y_i for the signer and use the private key sk_i to inversely compute x_i, i.e., $x_i = g^{-1}(y_i)$
- Return the signature $\sigma = (v, x_1, x_2, ...x_n)$

VERIFY(m, σ):

- Apply the hash function to the message m to obtain the key as in the sign function.
- Calculate all y_j values using the trapdoor function g.
- Verify whether the following equation holds: $v = C_{k,v}(y_1, y_2...y_i...y_n) = E_k(y_n \oplus E_k(y_{n-1} \oplus \dots (E_k(y_i) \oplus \dots E_k(y_1)\dots)))$

Shamir's ring signature scheme, as a pioneering work in the field, provides strong anonymity for signing within a given set of users and ensures unforgeability. However, it has some drawbacks:
Linearity: There is a potential attack in which an adversary can create a valid ring signature by combining existing valid signatures without knowing the corresponding secret keys. This is because Shamir's approach is based on linear operations.
Scalability: The size of the signature generated by this scheme is linear with the number of users in the group (O(n)). This can significantly impact performance in real-world scenarios, particularly for large user groups.

3.2 AOS Ring Signature

Building upon the foundation laid by Rivest, Shamir, and Tauman's original ring signature scheme, Abe, Okamoto, and Suzuki (AOS) introduced an improved ring signature scheme[1] that addresses some of the limitations of the original.

In the context of AOS Ring Signatures, the Schnorr signature[15] is used as a building block to construct a ring signature that maintains the anonymity of the signer within a group of possible signers.

Definition 1 (Schnorr Signature)**.** • **Key Generation**: The private key is a random number selected from a finite field, $sk \in F_p$. And the public key is generated as $pk = g^{sk}$.

- **Sign**: To sign a message m, the sender first generates a random number r (known as a nonce). Then, they compute a commitment R to the nonce as $R = g^r$. The sender then computes a hash of the message and the commitment, $e = H(m|R)$, multiplies this hash by the private key, and subtracts the result from the nonce to create the signature, $\sigma = r - e \cdot sk$.

- **Verify**: To verify the signature, the verifier raises the generator to the power of the signature and multiplies the public key by the hash, $e_v = g^{\sigma} \cdot pk^e$. Finally, checks whether $e_v = e$.

In AOS Ring Signatures, to sign a message, the actual signer (who is one of the potential signers) computes a series of Schnorr signatures and random values in a way that forms a closed loop (a "ring"), which hides the identity of the actual signer. Importantly, the computation involves a "trapdoor" that only the actual signer can navigate, which ensures that only a member of the group could have produced the signature. To verify the ring signature, any observer can check the consistency of the computed Schnorr signatures and random values across the entire loop. If they all "match up", the ring signature is valid. However, the observer cannot determine which of the potential signers was the actual signer, which preserves the anonymity of the signer. We conclude the AOS scheme as follows:
SIGN(m, PKs, sk):

- Signer i use a random filed element r to generate the commitment R_i as: $R_i = g^r$.

- Generate the challenge c_{i+1} for next user in the group by $c_{i+1} = H(m|R_i)$.

- Randomly generate the response z_j for all users except the signer i, forge the challenges for these users by $c_j = H(m|R')$, where $R' = g^{z_{j-1}} \cdot pk_{j-1}$.

- Compute z_i for signer by: $z_i = r - c_i \cdot sk_i$.

- Return the signature σ as: $(c_1, z_1, z_2, \ldots, z_n)$

VERIFY(m, σ):

- Calculate R_j based on all responses z in signature as: $R_j = g^{z_{j-1}} \cdot pk_{j-1}$

- check whether $H(m|R_1) = c_1$ holds.

Compared to Shamir's Ring Signature, the AOS Ring Signature does not possess the property of linearity. This means that one cannot construct a valid signature by adding multiple valid signatures together, making it relatively more secure.

Furthermore, AOS Ring Signature is compatible with the Elliptic Curve Digital Signature Algorithm (ECDSA)[6]. ECDSA is a digital signature scheme that uses elliptic curve cryptography (ECC) based on Elliptic Curve Discrete Logarithm Problem (ECDLP) to provide the same level of security as RSA but with significantly smaller keys.

Theorem 1 (Elliptic Curve Discrete Logarithm Problem (ECDLP)). *Consider two points P and Q that belong to the elliptic curve $E(F_q)$ such that P equals the result of Q multiplied by an integer a. Without prior knowledge of a, there exists no algorithm that can efficiently compute the value of a within a polynomial time. Essentially, it is computationally challenging to determine a solely from the given points P and Q.*

Symmetric security level(bytes)	RSA Key Size (bytes)	Elliptic Curve Key Size (bytes)
10	128	20
14	256	28
16	384	32
24	960	48
32	1920	64

Table 1: Comparison of key sizes for different security levels

Therefore, compared to Shamir's Ring Signature which uses RSA encryption, ECDSA requires a far smaller key size for the same security level. This makes AOS Ring Signature a more efficient choice for applications that require both security and efficiency.

However, the AOS Ring Signature doesn't address the limited scalability issue found in Shamir's Ring Signature. The size of the final generated signature in the AOS model is still linear with the size of the group. This characteristic significantly constrains its practical applications.

3.3 Linkable Ring Signature

Another issue to consider is that previous ring signature schemes lack linkability, which means they do not prevent a signer from generating multiple signatures using the same key. This property is critical for practical applications of ring signatures. For instance, in an e-voting system, we need to prevent double-voting problems, and in the context of cryptocurrencies, it is essential to avoid the double-spending issue.

Liu, et al, first proposed the linkable ring signature concept called Linkable Spontaneous Anonymous Group(LSAG) Signature [8], which allows the verifier to determine whether two signatures are signed by a same signer using the identical key, that is, whether the two signatures are linked. Importantly, this identification is accomplished without compromising the anonymity of the signer. In LSAG, we need to add a another generator h based on the public keys: $h = H_2(PKs)$, where $H_2()$ is a special Hash Function returns a group element. An Overview of LSAG is as followed:

SIGN(m, PKs, sk):

- Copy the public key of the signer i based on generator h: $pk'_i = h^{sk_i}$
- Randomly choose a scalar u, and forge the challenge for user i+1 by: $c_{i+1} = H(PKs, pk_i, m, g^u, h^u)$.
- Randomly generate the response z_j for all users except the signer i, forge the challenges for these users by $c_{j+1} = H(PKs, pk, m, g^{z_j}, pk_j^{c_j}, h^{z_j} pk_j'^{c} j)$.

- Compute z_i for signer by: $z_i = u - sk_i \cdot c_i \bmod q$.
- Return the signature σ as: $(c_1, z_1, z_2, \ldots, z_n, pk'_i)$

VERIFY(m, σ):

- For $i = 1, \ldots, n$, compute $c_{j+1} = H(PKs, pk, m, g^{z_j}, pk_j^{c_j}, h^{z_j} pk_j'^{c_j})$
- check whether $c_1 = H(PKs, pk, m, g^{z_n}, pk_n^{c_n}, h^{z_n} pk_n'^{c_n})$ holds.

Furthermore, Liu subsequently proposed a secure model for Linkable Ring Signatures[9]. Apart from the previously mentioned properties of completeness, robustness, and anonymity, this model also includes a property termed 'Linkability'. A ring signature with linkability requires a Link function, which assesses whether two signatures are generated by the same signer based on the same secret key, indicating that the two signatures are linked.

Definition 2 (LINK)**.** The function Link $\left(1^k, 1^n, PKs, m_1, m_2, \sigma_1, \sigma_2\right)$ is a boolean algorithm that receives as inputs: a security parameter k, n potential signers associated with n unique public keys PKs, two distinct messages m_1, m_2, and two signatures σ_1, σ_2. The signatures must satisfy the condition that Verify $\left(1^k, 1^n, PKs, m_1, \sigma_1\right) = 1$ and Verify $\left(1^k, 1^n, PKs, m_2, \sigma_2\right) = 1$. The Link function then returns 1 if the signatures are linked (i.e., signed by the same signer) and 0 if they are not.

$$\text{Link}\left(1^k, 1^n, PKs, m_1, m_2, \sigma_1, \sigma_2\right) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where i and j are the indexes of the signers for σ_1 and σ_2

Although the LSAG signature enhances the AOS ring signature by adding linkability to prevent the issue of double-signed signatures, it still fails to address the problem where the size of the ring signature is linearly related to the group size.

3.4 One-out-of-Many proofs

Groth and Kohlweiss[5] initiated the concept of One-out-of-Many proofs, which marked the transition of the ring signature size from linear to logarithmical. Bootle[3] et al., then further optimized this protocol by applying Pedersen vector commitment.

Definition 3 (Pedersen vector commitment)**.** Given a vector of values $(x_0, x_1 \ldots x_n)$, instead of commit the individual entry separately, one can commit the vector as a whole as:

$$c = Com(x_0, x_1 \ldots x_n) = g^{x_0} \cdot g^{x_0} \cdot \cdots \cdot g^{x_n} \cdot h^r$$

Where

- c is the commitment.
- $(g_1, g_2, \ldots, g_n)$are pre-selected group elements in a cryptographic group G (a set of elements with specific properties). These elements are selected once during the setup phase and are publicly known.

- $h$is another pre-selected group element in G, distinct from all the gs, and is also publicly known.

- r is a random number known as a blinding factor. To prevent using brute force attacks extract the vetor of witness.

The core idea behind One-out-of-Many proofs is as follows:
The prover, given a vector of commitments $(c_j)_{j=0}^{N-1}$, aims to demonstrate that it has knowledge of a commitment c_i that opens to Verifier at an index i within the set $0, ..., N-1$. This index i is uniquely known to the prover. The general concept entails proving that $\prod_{j=0}^{N-1} c_j^{\lambda_{ji}}$ is a commitment to zero, where $\lambda_{j\theta}$ is the Kronecker lambda function, with $\lambda_{i\theta} = 1$ and $\lambda_{ji} = 0$ for $j \neq i$. The prover then expresses the index i in binary form as $i = \sum_{l=0}^{L-1} 2^l i_l$ and provides proof that $\prod_{j=0}^{N-1} c_i^{\lambda_{j\theta}} = \prod_{j=0}^{N-1} c_j^{\prod_{l=0}^{L-1} \lambda_{j_l i_l}}$ is a commitment to zero. Here, l belongs to $0, ..., L-1$ and $L = \log N$. Then the prover must open a masking value $f_l = \theta_l e + m_l$ for each l, where m_l is a randomly selected value to ensure that f_l does not divulge any information about θ_l, and e is a random challenge issued by the verifier.

Bootle et al., utilizes the vector commitment to verify that $\theta_l \in 0, 1$ for $l \in 0, ..., L-1$:

$$\text{Commit}\left((f_l)_{l=0}^{L-1}\right) = \text{Commit}\left((\theta_l)_{l=0}^{L-1}\right)^e \cdot \text{Commit}\left((m_l)_{l=0}^{L-1}\right)$$

$$\prod_{l=0}^{L-1} g_l^{f_l(e-f_l)} = \text{Commit}\left(\left((1-2\theta_l)\, m_l\right)_{l=0}^{L-1}\right)^e \cdot \text{Commit}\left(\left(m_l^2\right)_{l=0}^{L-1}\right)$$

The first equation ensures unary linearity between θ_l and f_l, while the second equation confirms $\theta_l \in 0, 1$ by checking if quadratic terms e^2 are nullified.

The Ring Signature construction based on One-out-of-Many proofs are as follows:
SIGN(m, PKs, sk):

- Generate the Common Reference String(CRS) as the security parameter.

- Commit the public keys based on the secrete key : $c = Com_{CRS}(0, sk)$

- Apply One-out-of-Many prove a: $a = P(CRS, PKs, j, sk)$

- Compute x by: $x = H(CRS, m, PKs, a)$

- Apply One-out-of-Many prove on x as: $z = P(x)$

- return $\sigma = (a, z)$

VERIFY(m, σ):

- Reveal x based on a: $x = H(CRS, m, PKs, a)$

- Apply One-out-of-Many Verification on (CRS, PKs, a, x, z)

The innovative One-out-of-Many proof marks another significant milestone in the field of ring signatures since its inception, ingeniously combining digital signatures with zero-knowledge proof. This combination dramatically reduces the size of the ring signature from linear to logarithmic, signifying a remarkable achievement.

Despite the undeniable contributions and role of One-out-of-Many proofs in the domain of cryptography, certain limitations persist. Firstly, the process of generating One-out-of-Many signatures is not transparent, necessitating reliance on a trusted third party to generate a Common Reference String (CRS). However, in practice, a fully reliable third party is not a given, indicating that the security assurance of One-out-of-Many proofs can still be enhanced.

Furthermore, while the signature size of One-out-of-Many proofs is no longer proportional to the number of participants (O(n)), there remains a theoretical potential for further improvements in this area. The upcoming DualRing signature scheme will illustrate this point.

3.5 DualRing Signature

In 2018, Bootle et al[4]. introduced a new, transparent zero-knowledge proof scheme called Bulletproof. Similar to the Pedersen vector commitment, Bulletproof is also used to commit a vector of values. However, this scheme is transparent, meaning that the Prover and the Verifier can independently generate and verify proofs, without the need for an additional third party to generate security parameters in advance. Moreover, the size of the proof generated by this scheme is also logarithmic (log(n)). The completeness of Bulletproof lies in the validity of Equation (1).

$$
\begin{aligned}
&\langle w, \mathbf{g} \rangle + L + R \\
&\qquad = \langle w, \mathbf{g} \rangle + r^2 \langle w_L, \mathbf{g}_R \rangle + r^{-2} \langle w_R, \mathbf{g}_L \rangle \\
&\qquad = (\langle w_L, \mathbf{g}_L \rangle + \langle w_R, \mathbf{g}_R \rangle) + r^2 \langle w_L, \mathbf{g}_R \rangle + r^{-2} \langle w_R, \mathbf{g}_L \rangle \\
&\qquad = \langle r w_L, r^{-1} \mathbf{g}_L \rangle + \langle r^{-1} w_R, r \mathbf{g}_R \rangle + \langle r w_L, r \mathbf{g}_R \rangle + \langle r^{-1} w_R, r^{-1} \mathbf{g}_L \rangle \\
&\qquad = \langle r w_L + r^{-1} w_R, r^{-1} \mathbf{g}_L + r \mathbf{g}_R \rangle \\
&\qquad = \langle w', \mathbf{g}' \rangle
\end{aligned}
\tag{1}
$$

Here are the key terms used in the above equation:

- r is a scalar, chosen randomly.
- The witness vector is represented as w while $\mathbf{g}$ represents the generator vector.
- $\langle w, \mathbf{g} \rangle$ is the inner product of vectors w and $\mathbf{g}$, denoted as $\langle w, \mathbf{g} \rangle = \sum_{i=1}^{n} w_i \cdot \mathbf{g}_i$
- w_L and w_R refer to the left half and right half of vector w, respectively. Similarly, $\mathbf{g}_R$ and $\mathbf{g}_L$ denote the right and left halves of the generator vector.

Initially, the prover pass the inner product of vectors u and v to Verifier as commitment. Following this, the prover consistently appends $r^2 \langle w_L, \mathbf{g}_R \rangle$ and $r^{-2} \langle w_R, \mathbf{g}_L \rangle$ to the proof, utilizing w' and g' as vectors w and g for the upcoming round. At the end, vectors w and g will be reduced to a single element where is the base case, and the final inner product of w

and g will be added to the proof. The entire process involves halving the input log(n) times. Consequently, the proof contains log(n) pairs of (L, R) values, making the total proof size 2log(n).

For the verifier's role, it commences with the Commitment C and employs the L and R values in the proof to generate the same r value as the prover in each iteration. They also calculate the new w' and g'values. However, as the verifier doesn't know the value of the given coefficient ci, they can't independently compute w'. Eventually, the verifier evaluates whether $w^* \cdot g^* = C$ can be established.

Yuen and his team[16] ingeniously developed a new Schnorr-based ring signature scheme, DualRing, in 2021. This was combined with BulletProof, resulting in a transparent ring signature with log(n) size.

The construction of DualRing is very similar to the traditional AOS ring signature. However, to make the new ring signature compatible with BulletProof, instead of containing one challenge and n responses like the AOS ring signature, DualRing consists of n challenges and one response. The specific algorithm for DualRing is as follows:
SIGN(m, PKs, sk):

- Select a random number r
- Forge n-1 challenges c_j for all other users.
- Accumulate the commitment by $A = r \cdot G_0 + \sum_{j \neq i} c_j \cdot pk_j$
- Calculate the challenge for signer i: $c_i = H(m, \mathbf{pks}, A) - \sum_{j \neq i} c_j$
- Generate the response $z = r - c_i \cdot sk_i \mod p$.
- Return the signature σ: $(z, c_1, c_2, \ldots, c_n)$

VERIFY(m, σ):

- Reconstruct the Commitment: $A = r \cdot G_0 + \sum_j^n c_j \cdot pk_j$.
- Sum the challenges $c = \sum_j^n c_j$.
- Check whether $c = H(m, \mathbf{pks}, A)$

In simple terms, a dual ring signature is constructed by fabricating n-1 commitments (c_j) for the other n-1 users. Then, based on these commitments, the commitment c_j for the signer j is created. These commitments together constitute a dual ring signature. If Prover does not know one of the secret key sk_i, the only way for Prover pass the verification is find a z where it satisfies $\sum_j c_j = H(m, pks, zG0 + \sum_j c_j \cdot pk_j)$. But according to the Hash function's security property, it is impossible to find such z within polynomial time.

Next is to Integrate the Dualring with BulletProof. Yuen regard the series of challenges c as the vector of scalars (witness) w in Inner Product Arguments, and consider all the users' public keys as the vector of generators (g), Then apply Inner Product Arguments to ultimately generate log(n) pairs of (L, R) as the siganture. In this case, the final size of DualRing signature is 2log(n)+4.

As of the present, DualRing represents the most compact and efficient transparent ring signature scheme. However, the prospect of further reducing the size of the signature, without impacting its transparency, remains an open area for investigation. This potential for improvement indicates a continuing need for research and innovation in the field of cryptographic signatures.

4 Applications

4.1 Confidiential Transaction

Confidential Transactions (CT)[10] is a concept proposed by Gregory Maxwell, aimed at enhancing the privacy and security of Bitcoin transactions. It's an upgrade to the protocol that can be added to Bitcoin to hide specific transaction amounts, thereby improving its privacy features. Privacy of transaction amounts is crucial. Analysis of public transaction amounts could potentially reveal the identities of transacting parties, especially in an open and transparent environment like blockchain. Additionally, transaction data could be maliciously exploited, such as for market manipulation.

Confidential Transactions work by encrypting the values of Bitcoin transaction outputs, which are usually visible to the public on the blockchain. This encryption allows only the sender and receiver to know the exact amount of the transaction. While the actual amount transferred is hidden, CT still allows network nodes to verify that a transaction balances (i.e., the total value of inputs equals the total value of outputs) without knowing the actual amounts. This is achieved through cryptographic techniques involving homomorphic encryption, specifically the Pedersen Commitment Scheme.

However, in the CT protocol, the addresses of the sender and receiver are not hidden, CT merely obscures the amount being transacted. If attackers are aware of a user's transaction address, they may attempt what is known as a "dust attack" or other types of attacks to try to gather more information about the user or directly steal funds. We refer readers to "An Analysis of Anonymity in the Bitcoin System"[12] for an understanding of attacks related to addresses.

For complete privacy, Shen Noether[11] proposed Ring Confidential Transactions (Ring CT), initially to be able to hide transaction amounts like CT. Ring CT uses a mathematical concept called Pedersen Commitments. This technique allows the hiding of transaction values by encrypting them but still enabling the network to verify if the total inputs and total outputs of a transaction match, ensuring no Monero coins are created out of thin air.

Moreover, Ring CT builds on LSAG to create Multilayered Linkable Spontaneous Ad-Hoc Group Signatures. Unlike LSAG, to better integrate with the CT protocol, MLSAG is designed to handle sets of key-vectors rather than just sets of individual keys. By using MLSAG, this allows the identity of the sender to be hidden among a set of other possible senders. By combining these techniques, Monero ensures the privacy of sender, receiver, and transaction amount, thus providing a high level of transaction privacy.

However, when Ring CT was first introduced, the size of ring signatures was linear. Therefore, when the One-out-of-Many proofs, a log size ring signature, was introduced, Russell[7] proposed a new Ring CT scheme called Omniring. Omniring uses One-out-of-Many

proofs to propose an upgraded version of the MLSAG (Multilayered Linkable Spontaneous Anonymous Group) signature scheme, called the "Omnisig" scheme, which allows it to scale logarithmically with the number of decoy outputs. This results in smaller transaction sizes and quicker verification times, thus improving the overall efficiency and scalability of the system.

5 Current Gaps and Future Work

Ring signature research primarily revolves around three central trajectories:

Improvements in Ring Signature Construction:

Reduction in Signature Size: The most compact ring signature today has a size of 2log(n)+4. Although not linearly related to the group size, this can be impractical for significantly large groups. Thus, it would be interesting to see if a new scheme could further reduce the signature size while maintaining transparency (that is, without involving a trusted third-party setup). Ideally, this scheme should generate a constant size signature.

Decrease in Verification Time: In most ring signature applications, the signers are distributed, and the verifier is centralized. This scenario often results in verification becoming a time bottleneck. As such, efforts are being made to minimize verification time, even if it means somewhat increasing the signing time. Of course, it would be ideal if all runtime components could be reduced.

Advancements in Application Domains:

The challenge lies in the practical integration of cutting-edge ring signature schemes. New ring signature schemes have been proposed in recent years, but a gap persists between these and the ones deployed in real-world applications. For instance, the ring signature scheme employed in latest Ring Confidential Transactions (Ring CT) still uses the out-of-Many proofs concept. Could we possibly devise a more efficient Ring CT scheme based on the currently most compact DualRing? To make this possible, the existing signature scheme might need modifications for compatibility, posing a substantial challenge in this direction.

Security-related Research:

With the rapid progress in quantum computing, traditional cryptographic algorithms are facing significant challenges because their security largely relies on the computational infeasibility of certain problems such as integer factorization and discrete logarithm problems, which quantum computers are predicted to solve efficiently. Although the current computational power of quantum computing is not yet strong enough to disrupt existing encryption systems, it's essential to take preventive measures. The proposition of a quantum-resistant ring signature scheme is a major hot topic. This will not only ensure protection in the advent of malicious use of quantum computing but also facilitate a timely technological transition towards quantum resistance.

References

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In Yuliang Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, pages 415–432, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[2] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 60–79, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[3] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on ddh. Cryptology ePrint Archive, Paper 2015/643, 2015. `https://eprint.iacr.org/2015/643`.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, 2018.

[5] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 253–280, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[6] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *Int. J. Inf. Secur.*, 1(1):36–63, aug 2001.

[7] Russell W. F. Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling private payments without trusted setup. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 31–48, New York, NY, USA, 2019. Association for Computing Machinery.

[8] Joseph Liu, Victor Wei, and Duncan Wong. Linkable spontaneous anonymous group signature for ad hoc groups. volume 2004, page 27, 07 2004.

[9] Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 614–623, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[10] Gregory Maxwell. Confidential transactions, 2015.

[11] Shen Noether. Ring confidential transactions. 2016.

[12] F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Privacy, security, risk and trust (passat), 2013 ieee third international conference on social computing (socialcom)*, pages 1318–1326. IEEE, 2013.

[13] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, 1978.

[14] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[15] C. P. Schnorr. Efficient signature generation by smart cards. *J. Cryptol.*, 4(3):161–174, jan 1991.

[16] Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. Dualring: Generic construction of ring signatures with efficient instantiations. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 251–281, Cham, 2021. Springer International Publishing.