mathematics

Article

A Universally Composable Linkable Ring Signature Supporting Stealth Addresses

Xingkai Wang [1], Chunping Zhu [1] and Zhen Liu [1,2,*]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; starshine87@sjtu.edu.cn (X.W.); chengfengpolang@sjtu.edu.cn (C.Z.)
[2] Shanghai Qizhi Institute, Shanghai 200003, China
* Correspondence: liuzhen@sjtu.edu.cn

Abstract: The linkable ring signature supporting stealth addresses (SALRS) is a recently proposed cryptographic primitive, which is designed to comprehensively address the soundness and privacy requirements associated with concealing the identities of both the payer and payee in cryptocurrency transactions. However, concerns regarding the scalability of SALRS have been underexplored. This becomes notably pertinent in intricate blockchain systems where multiple cryptographic primitives operate concurrently. To bridge this gap, our work revisited and formalized the ideal functionality of SALRS within the universal composability (UC) model. This encapsulates all correctness, soundness, and privacy considerations. Moreover, we established that the newly proposed UC-security property for SALRS is equivalent to the concurrent satisfaction of signer-unlinkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability. These properties represent the four crucial game-based security aspects of SALRS. This result ensures the ongoing security of previously presented SALRS constructions within the UC framework. It also underscores their adaptability for seamless integration with other UC-secure primitives in complex blockchain systems.

Keywords: universal composability; signature scheme; stealth address; privacy preserving

PACS: J0101

MSC: 94A62

check for updates

Citation: Wang, X.; Zhu, C.; Liu, Z. A Universally Composable Linkable Ring Signature Supporting Stealth Addresses. *Mathematics* **2024**, *12*, 491. https://doi.org/10.3390/math12030491

Academic Editor: Antanas Cenys

Received: 21 November 2023
Revised: 30 December 2023
Accepted: 23 January 2024
Published: 3 February 2024

1. Introduction

In traditional cryptocurrencies such as Bitcoin and Ethereum, the anonymity they provide is at a pseudonymous level. During transactions, it is not possible to link the wallet address to the real identity of the transactor. However, privacy-focused cryptocurrencies like Monero or Zcash demand the preservation of both payer and payee anonymity and unlinkability in transactions. In some of the blockchain systems, e.g., CryptoNote [1], linkable ring signatures (LRS) [2] and the key derivation mechanism (KeyDerM) [1] are employed to address the aforementioned goals of anonymity and unlinkability.

Specifically, when a payer intends to conduct a payment transaction with a payee, the payer first utilizes KeyDerM to derive a derived public key from the payee's master public key as the receiving address for the transaction. As the payee's master public key does not appear in the transaction, the recipient of this transaction, i.e., the payee, cannot be identified. KeyDerM is also known as the stealth address (SA) [3] mechanism. When the payee wishes to spend the currency associated with this derived public key, they need to select a ring of derived public keys during the transaction. This ring includes their own derived public key. Through this ring, a linkable ring signature is generated, allowing anyone to verify the validity of the signature without knowing the actual signer. The linkability aspect is also useful in detecting double-spending behavior by the signer, as two different signatures generated for the same derived public key will be linked.

Recently, there has been significant attention in the community on linkable ring signatures (LRS) and stealth addresses (SA) [4–8]. For instance, in projects like Monero [9] and CryptoNote [1], LRSs and KeyDerM are considered foundational constructs, but they are treated as separate entities without a unified security analysis, despite their tight coupling in usage. The existing literature [2,10–12] largely addresses LRSs or SAs individually, particularly in the context of standard signature schemes [4,8]. Moreover, the signature keys and public keys used in LRSs are generated by the SA mechanism, which means that the LRS mechanism used in the blockchain system does not independently generate keys. Further research is needed to explore the security and privacy aspects of key generation in SA. Whether the security and privacy models of linkable ring signatures and stealth addresses can be effectively applied in cryptocurrency scenarios requires thorough analysis by researchers. This is especially pertinent in the context of key selection attacks by adversaries, where existing linkable models either lack consideration for such attacks or fail to align with the practical use cases of cryptocurrencies.

In order to address the aforementioned issues, Liu et al. [13] proposed a new cryptographic primitive, namely the linkable ring signature supporting stealth addresses (SALRS). This scheme aims to fulfill the security and privacy requirements of concealing both the payer and the payee in cryptocurrency transactions. The security model of SALRS provides properties such as strong unforgeability, signer-linkability, and signer-non-slanderability. The privacy model ensures properties like signer-anonymity, master-public-key-unlinkability, and derived-public-key-unlinkability. All these properties can be concurrently defined in the SALRS model, aligning with the practical requirements of cryptocurrency scenarios, especially in the context of key selection attacks. Liu et al. [13] also introduced a lattice-based construction for SALRS and demonstrated its privacy and security under the random oracle model. However, there has not been dedicated research on the universal composability (UC) of SALRS to date. This section will analyze and study the UC security of SALRS, providing separate proofs for its security and privacy under UC security definitions. The conclusion drawn will affirm that SALRS satisfies UC security, enhancing its security and practicality in application scenarios like cryptocurrency.

1.1. Our Results

In this paper, we revisit the security definition of SALRS and explore its modularity and adaptability to other cryptographic primitives within a comprehensive cryptocurrency system. Our contributions can be summarized as follows.

- We provide a novel security definition of linkable ring signatures supporting stealth addresses (SALRS) in the universal composability (UC) framework. We define the ideal functionality, which simultaneously captures correctness, signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability. This is a more robust simulation-based security definition, implying that the protocol remains secure even when composed with arbitrary protocols.
- We further investigate the security level of the proposed security definition. Through rigorous analysis, we demonstrate that the proposed UC-security of SALRS is equivalent to the concurrent satisfaction of signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key unlinkability.
- We establish that the ideal functionality can be securely realized by the previously proposed construction that achieving the former four security definitions. This finding indicates that, including the SALRS construction proposed in [13], all secure SALRS constructions satisfy the security definition of [13], are UC-secure, and can arbitrarily compose with other UC-secure components in a complicated blockchain system.

1.2. Related Work

Before Liu et al. [13] gave the first practical quantum-resistant solution that hides the payers and payees of transactions in cryptocurrencies, there were several studies on linkable ring signatures [5,14–16], but none of them introduced stealth addresses. Without taking efficiency into account, [17,18] can also attain a logarithmic signature size concerning the number of signers in the ring. The constructions supporting stealth addresses [4,8] do not fulfill the criteria for linkable ring signature satisfaction.

While our work is the first to specifically address the UC-security of SALRS, it is worth noting that there have been various studies focusing on UC-secure signature schemes. Canetti [19] initially proposed a functionality for signature schemes, but a flaw in the definition made secure realization impossible. Subsequently, Backes et al. [20] and Canetti [21] addressed the flaw, establishing that the newly defined UC-security is equivalent to the game-based definition of EUF-CMA. In this paper, we employ a similar proven technique to circumvent the flaw identified in [19]. Apart from typical signature schemes, Abe et al. [22] introduced the UC-secure non-committing blind signature. Later, Hong et al. [23] formally defined the UC security of proxy re-signature. More recently, Zhu et al. [24] discussed the UC-security of the key-insulated and privacy-preserving signature scheme with publicly derived public key (PDPKS). While similar techniques are employed in defining the ideal functionality of digital signatures, it is crucial to emphasize that SALRS is distinct from these signature-related primitives, offering unique functionality and security features.

1.3. Outline

In Section 2, we show the syntax and security definitions of the primitive linkable ring signature with stealth addresses (SALRS), and preliminaries on the universal composability framework. In Section 3, we define the ideal functionality of SALRS, which captures its UC-security. In Section 4, we prove the existence of a UC-secure construction, by proving the equivalence between the game-based security [13] and the newly defined security. This paper is concluded in Section 5.

2. Preliminaries

In this section, we begin by revisiting the definition of SALRS as proposed by Liu et al. [13]. Next, we review the background of the Universal Composability (UC) framework [19], as well as the definition of UC-security.

2.1. SALRS: Linkable Ring Signature Supporting Stealth Addresses

2.1.1. Syntax

An SALRS scheme [13] consists of the following eight algorithms:

- $\mathsf{Setup}(\kappa) \to \mathsf{PP}$. Taking as input a security parameter κ, the algorithm outputs the system public parameter PP, which corresponds to the common parameters in the system.
- $\mathsf{MasterKeyGen}() \to (\mathsf{MPK}, \mathsf{MSK})$. Each user executes the master-key-generating algorithm to generate its master public–private key pair.
- $\mathsf{DPKDerive}(\mathsf{MPK}) \to \mathsf{DPK}$. Anyone can execute the derived public-key-generating algorithm to generate a fresh derived public key DPK from a master public key MPK.
- $\mathsf{DPKOwnerCheck}(\mathsf{DPK}, \mathsf{MPK}, \mathsf{MSK}) \to 0\backslash 1$. Taking as input a derived public key DPK and a master public–private key pair $(\mathsf{MPK}, \mathsf{MSK})$, the owner of the master public key can execute the derived public key owner checking algorithm to obtain a bit $b \in \{0,1\}$, indicating whether a derived public key DPK is a valid derived public key generated from its master public key MPK.
- $\mathsf{DPKPublicCheck}(\mathsf{DPK}) \to 0\backslash 1$. Taking as input a derived public key DPK, anyone can execute the derived-public-key-checking algorithm to obtain a bit $b \in \{0,1\}$, indicating whether the derived public key is well formed, so that it can use them as ring numbers for its ring signature generation.

- $\mathsf{Sign}(M, R, \mathsf{DPK}, \mathsf{MPK}, \mathsf{MSK}) \to \sigma$. Taking as input a message M, a ring of well-formed derived public keys $R = (\mathsf{DPK}_1, \ldots, \mathsf{DPK}_r)$, a derived public key $\mathsf{DPK} \in R$, and its corresponding master public-private key pair $(\mathsf{MPK}, \mathsf{MSK})$, the key owner can execute the signing algorithm to generate a signature σ on the message M with respect to the ring R.
- $\mathsf{Verify}(M, R, \sigma) \to 0\backslash 1$. Taking as input a message M, a ring of well-formed derived public keys R, and a purported signature σ on the message M with respect to the ring R, anyone can execute the verifying algorithm to obtain a bit $b \in \{0,1\}$ indicating the validity of the signature.
- $\mathsf{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) \to 0\backslash 1$. Taking as input two valid signatures (M_0, R_0, σ_0) and (M_1, R_1, σ_1), anyone can execute the linking algorithm to obtain a bit $b \in \{0,1\}$ indicating whether two signatures are linked or unlinked.

Remark 1. *We consider a public key ring R as an ordered set. Specifically, it is composed of a set of public keys, and during the execution of* Sign() *and* Verify()*, the public keys are arranged in a specific order, each assigned a unique index.*

Remark 2. *We note that the nature of whether* Sign() *is probabilistic or deterministic remains open, as it may vary depending on the specific constructions employed.*

Correct. An SALRS scheme is correct if it satisfies the following property:
Let $\mathsf{PP} \leftarrow \mathsf{Setup}(\kappa)$,

- $\forall(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{MasterKeyGen}()$, $\mathsf{DPK} \leftarrow \mathsf{DPKDerive}(\mathsf{MPK})$, it holds that $\mathsf{DPKOwnerCheck}(\mathsf{DPK}, \mathsf{MPK}, \mathsf{MSK}) = 1$ and $\mathsf{DPKPublicCheck}(\mathsf{DPK}) = 1$.
- $\forall M \in \mathcal{M}$, any ring of well-formed derived public keys R, and $\forall \mathsf{DPK}_s \in R$ s.t. $\mathsf{DPKOwnerCheck}\ (\mathsf{DPK}_s, \mathsf{MPK}, \mathsf{MSK}) = 1$ for some master key pair $(\mathsf{MPK}, \mathsf{MSK})$, it holds that $\mathsf{Verify}(M, R, \mathsf{Sign}\ (M, R, \mathsf{DPK}_s, \mathsf{MPK}, \mathsf{MSK})) = 1$.
- $\forall M_0, M_1 \in \mathcal{M}$, any well-formed derived public key rings R_0, R_1, and $\forall \mathsf{DPK}_{s_0} \in R_0$, $\mathsf{DPK}_{s_1} \in R_1$, s.t. $\mathsf{DPKOwnerCheck}(\mathsf{DPK}_{s_i}, \mathsf{MPK}_i, \mathsf{MSK}_i) = 1$ for some master key pairs $(\mathsf{MPK}_i, \mathsf{MSK}_i)$ $(i \in \{0,1\})$, $\sigma_i \leftarrow \mathsf{Sign}(M_i, R_i, \mathsf{DPK}_{s_i}, \mathsf{MPK}_i, \mathsf{MSK}_i)(i \in \{0,1\})$, it holds that
$$\Pr[\mathsf{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 1] = 1 \tag{1}$$
if $\mathsf{DPK}_{s_0} = \mathsf{DPK}_{s_1}$, and
$$\Pr[\mathsf{Link}(M_0, R_0, \sigma_0, M_1, R_1, \sigma_1) = 1] \geq 1 - negl(\kappa) \tag{2}$$
if $\mathsf{DPK}_{s_0} \neq \mathsf{DPK}_{s_1}$.

2.1.2. Security Models

Below, we provide the security definitions of SALRS, including soundness and privacy. Specifically, soundness encompasses unforgeability, signer-linkability, and signer-non-slanderability, while privacy includes signer-anonymity, master-public-key-unlinkability, and derived-public-key-unlinkability [13].

In more detail, *unforgeability* holds when only the user possessing the secret key for some public key in a ring can generate a valid signature with respect to that ring. *Signer-linkability* concerns the scenario where, with respect to a derived public key, if the key owner generates two or more valid signatures, these signatures will be identified as linked. This fulfills the security requirement of preventing double spending in cryptocurrencies. *Signer-non-slanderability* ensures that no one can falsely implicate other users by creating a signature linked to the signature of the target user.

For privacy requirements, *signer-anonymity* ensures that, given a valid signature for a ring of derived public key, it is infeasible for anyone to identify the signer's derived public key within the ring. This property captures the privacy-preserving requirement of concealing the payer's identity. *Master-public-key-unlinkability* ensures that, given a derived public key and its corresponding signatures, it is impossible to determine which master public key, from a known set of master public keys, was the origin of the derivation. *Derived-public-key-unlinkability* ensures that, given two derived public keys and their corresponding signatures, it is impossible to ascertain whether they are derived from the same master public key. This property ensures privacy by obscuring the link between payees in different transactions.

Particularly, Liu et al. [13] shows that unforgeability can be implied from signer-linkability and signer-non-slanderability together, and derived public-key-unlinkability can be implied from master public-key-unlinkability. We focus mainly on the remaining four properties in this paper. Formal definitions on the security properties are shown as follows.

Definition 1 (Signer-Linkability)**.** *For an SALRS scheme defined according to the specifications described above, for any PPT adversary $\mathcal{A}$, consider the following experiment $\textbf{\textit{Exp}}_{\mathcal{A}}^{\mathsf{snlink}}(\kappa)$:*

- ***Setup Phase.*** *$\mathsf{PP} \leftarrow \mathsf{Setup}(\kappa; r)$ is executed, where r represents the randomness used within $\mathsf{Setup}()$. $\mathcal{A}$ acquires both PP and r.*
- ***Output Phase.*** *The adversary $\mathcal{A}$ outputs a set of tuples $\{(M_i^*, R_i^*, \sigma_i^*)\}_{i \in [k]}$, where $k \geq 2$.*

The adversary $\mathcal{A}$ succeeds if (1) $\forall i \in [k]$, it holds that $\mathsf{Verify}(M_i^*, R_i^*, \sigma_i^*) = 1$, (2) $\forall i, j \in [k]$, $i \neq j$, $\mathsf{Link}(M_i^*, R_i^*, \sigma_i^*, M_j^*, R_j^*, \sigma_j^*) = 0$, and (3) $| \cup_{i \in [k]} R_i^* | < k$.

The SALRS scheme is signer-linkable, if for any PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that $\Pr[\mathcal{A}$ succeeds in $\textbf{\textit{Exp}}_{\mathcal{A}}^{\mathsf{snlink}}(\kappa)] \leq \mathsf{negl}(\cdot)$.

Definition 2 (Signer-Non-Slanderability)**.** *For an SALRS scheme defined according to the specifications described above, for any PPT adversary $\mathcal{A}$, consider the following experiment $\textbf{\textit{Exp}}_{\mathcal{A}}^{\mathsf{snnsl}}(\kappa)$:*

1. ***Setup Phase.*** *$\mathsf{PP} \leftarrow \mathsf{Setup}(\kappa; r)$ is executed, where r represents the randomness used within $\mathsf{Setup}()$. $\mathcal{A}$ acquires both PP and r.*
   A set of master key generating algorithms $\{(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{MasterKeyGen}()\}_{i \in [poly(\kappa)]}$ are initiated, and the resulting set $\{\mathsf{MPK}_i\}_{i \in [poly(\kappa)]}$ is presented to $\mathcal{A}$.
   An empty set, $L_{dpk} = \emptyset$, is initialized, which serves the purpose of storing valid derived public keys derived from the target master public keys.
2. ***Probing Phase.*** *The adversary $\mathcal{A}$ can query the following two oracles adaptively:*
   - *Derived Public Key Adding Oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot, \cdot)$:*
     Taking as input a derived public key DPK and a master public key MPK_i, the adversary $\mathcal{A}$ receives from this oracle a bit $b \leftarrow \mathsf{DPKOwnerCheck}(\mathsf{DPK}, \mathsf{MPK}_i, \mathsf{MSK}_i)$. If the response $b = 1$, update $L_{dpk} = L_{dpk} \cup \{\mathsf{DPK}\}$.
   - *Signing Oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot, \cdot)$:*
     Taking as input a message $M \in \mathcal{M}$, a ring of well-formed derived public keys R, and a derived public key $\mathsf{DPK} \in R \cap_{dpk}$, the adversary $\mathcal{A}$ receives from this oracle a signature $\sigma \leftarrow \mathsf{Sign}(M, R, \mathsf{DPK}, \mathsf{MPK}_i, \mathsf{MSK}_i)$, where $(\mathsf{MPK}_i, \mathsf{MSK}_i)$ represents the master key pair for DPK.
3. ***Output Phase.*** *The adversary $\mathcal{A}$ outputs two well-formed tuples, denoted as $(\hat{M}, \hat{R}, \hat{\sigma})$ and (M^*, R^*, σ^*).*

Let $S_{so} = \{(M, R, \mathsf{DPK}, \sigma)\}$ be the query-answer tuples for $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot, \cdot)$. $\mathcal{A}$ succeeds if (1) $\mathsf{Verify}(M^*, R^*, \sigma^*) = 1$, (2) $(\hat{M}, \hat{R}, \hat{\mathsf{DPK}}, \hat{\sigma}) \in S_{so}$ for some $\hat{\mathsf{DPK}} \in \hat{R} \cap L_{\mathsf{DPK}}$, (3) $(M^*, R^*, \hat{\mathsf{DPK}}, \sigma^*) \notin S_{so}$, and (4) $\mathsf{Link}(M^*, R^*, \sigma^*, M, \hat{R}, \hat{\sigma}) = 1$.

The SALRS scheme is signer-non-slanderable if, for any PPT adversary $\mathcal{A}$, there is a negligible function $negl(\cdot)$ such that $\Pr[\mathcal{A}$ succeeds in $\boldsymbol{Exp}_{\mathcal{A}}^{snnsl}(\kappa)] \leq negl(\kappa)$.

Definition 3 (Signer-Anonymity)**.** *For an SALRS scheme defined according to the specifications described above, for any PPT adversary $\mathcal{A}$, consider the following experiment $\boldsymbol{Exp}_{\mathcal{A}}^{snano}(\kappa)$:*

- ***Setup Phase.*** *Same as the **Setup** phase in the experiment $\boldsymbol{Exp}_{\mathcal{A}}^{snnsl}(\kappa)$ as defined in Definition 2.*
- ***Probing Phase 1.*** *Same as the **Probing** phase in the experiment $\boldsymbol{Exp}_{\mathcal{A}}^{snnsl}(\kappa)$ as defined in Definition 2.*
- ***Challenge Phase.*** *The adversary $\mathcal{A}$ outputs a message $M^* \in \mathcal{M}$, a ring of well-formed derived public keys R^*, and two indices $i_0, i_1 \in [poly(\kappa)]$, such that (1) $i_0 \neq i_1$, (2) DPK_{i_0}, $DPK_{i_1} \in R^* \cap L_{dpk}$, and (3) none of DPK_{i_0} or DPK_{i_1} were queried as input of $\mathcal{O}^{Sign}$.*
  A challenge bit $b \in \{0,1\}$ is selected; the adversary $\mathcal{A}$ is provided with the signature $\sigma \leftarrow Sign(M^*, R^*, DPK_{i_b}, MPK, MSK)$, where (MPK, MSK) represents the master key pair for DPK_{i_b}.
- ***Probing Phase 2.*** *Same as **Probing Phase 1**, with the added condition that none of DPK_{i_0} or DPK_{i_1} were queried as an input of $\mathcal{O}^{Sign}$.*
- ***Output Phase.*** *The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as its guess for b.*

The advantage of the adversary $\mathcal{A}$ winning $\boldsymbol{Exp}_{\mathcal{A}}^{snano}(\kappa)$ is $\boldsymbol{Adv}_{\mathcal{A}}^{snano} = |\Pr[b \neq b'] - \frac{1}{2}|$.

The SALRS scheme is signer-anonymous if, for any PPT adversary $\mathcal{A}$, there is a negligible function $negl(\cdot)$ such that $\boldsymbol{Adv}_{\mathcal{A}}^{snano} \leq negl(\cdot)$.

Definition 4 (Master-Public-Key-Unlinkability)**.** *For an SALRS scheme defined according to the specifications described above, for any PPT adversary $\mathcal{A}$, consider the following experiment $\boldsymbol{Exp}_{\mathcal{A}}^{mpkunl}(\kappa)$:*

- ***Setup Phase.*** *Same as the **Setup Phase** in the experiment $\boldsymbol{Exp}_{\mathcal{A}}^{snnsl}(\kappa)$ as defined in Definition 2.*
- ***Probing Phase 1.*** *Same as the **Probing** phase in the experiment $\boldsymbol{Exp}_{\mathcal{A}}^{snnsl}(\kappa)$ as defined in Definition 2.*
- ***Challenge Phase.*** *The adversary $\mathcal{A}$ outputs two indices $i_0, i_1 \in [poly(\kappa)]$, such that $i_0 \neq i_1$. A challenge bit $b \in \{0,1\}$ is selected, and the adversary $\mathcal{A}$ is provided with the derived public key $DPK^* \leftarrow DPKDerive(MPK_{i_b})$. Update $L_{dpk} = L_{dpk} \cup \{DPK^*\}$.*
- ***Probing Phase 2.*** *Same as **Probing Phase 1**, with the added condition that none of $(DPK^*, MPK_{i_j})_{j\in\{0,1\}}$ were queried as an input of $\mathcal{O}^{DPKAdd}$.*
- ***Output Phase.*** *The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as its guess for b.*

The advantage of the adversary $\mathcal{A}$ winning $\boldsymbol{Exp}_{\mathcal{A}}^{mpkunl}(\kappa)$ is $\boldsymbol{Adv}_{\mathcal{A}}^{mpkunl} = |\Pr[b \neq b'] - \frac{1}{2}|$.

The SALRS scheme is master-public-key-unlinkable if, for any PPT adversary $\mathcal{A}$, there is a negligible function $negl(\cdot)$, such that $\boldsymbol{Adv}_{\mathcal{A}}^{mpkunl} \leq negl(\cdot)$.

With these comprehensive security and privacy models, SALRS effectively addresses the security- and privacy-preserving requirements essential in practical cryptocurrency scenarios. Notably, SALRS accommodates rings containing derived public keys that an adversary generated from their own master public keys. This realistic feature acknowledges situations where an attacker might create derived public keys from their master public keys, engaging in transactions among these keys with the intention of executing attacks, such as double spending or compromising the security and privacy of other users.

2.2. Universal Composability

We adopt the concept of universally composable security as defined by Canetti [19]. This framework offers a systematic approach to defining the security properties of cryptographic primitives, ensuring security is preserved under a general composition with an unbounded number of instances of arbitrary protocols running concurrently. Within this framework, all protocols operate in a specified computational environment in the presence of an adversary. The computational environment represents other protocols that may be concurrently executed alongside the protocol under consideration.

Given that communication is public, with no assurance of message delivery and is asynchronous without a guarantee of messages being delivered in order in the actual network, we presume that the communication between parties is authenticated. This authentication ensures that messages sent by honest parties will not be tampered with. We proceed by providing an overview of the model for protocol execution, known as the real-world model of computation. Subsequently, we introduce the ideal-world model of computation and present the general definition of security that realizes an ideal functionality.

In the real world, there exists an adversary A and a protocol π that realizes a functionality among several parties. We denote the output of environment Z when interacting with adversary A and parties $P_1, \ldots, P_n$ running protocol π on a security parameter k, auxiliary input z, and random input $r = (r_Z, r_A, r_1, \ldots, r_n)$, where each element represents the random tape used by the corresponding participant. We use the notation $\mathsf{REAL}_{\pi,A,Z}(k, z, r)$ to represent this output. Additionally, let $\mathsf{REAL}_{\pi,A,Z}(k, z)$ denote the random variable describing $\mathsf{REAL}_{\pi,A,Z}(k, z, r)$ when r is uniformly chosen.

In the ideal world, there is a simulator S that simulates the real-life scenario, an ideal functionality F, and n dummy parties for the integrity of the simulation. Let $\mathsf{IDEAL}_{F,S,Z}(k, z, r)$ denote the output of environment Z when interacting with adversary S and ideal functionality F on security parameter k, auxiliary input z, and random input $r = (r_Z, r_S, r_F)$, where each element represents the random tape used by the corresponding participants. Let $\mathsf{IDEAL}_{F,S,Z}(k, z)$ denote the random variable describing $\mathsf{IDEAL}_{F,S,Z}(k, z, r)$ when r is uniformly chosen.

The definition of universal composability is shown as follows.

Definition 5 (Universal Composability [19])**.** *A protocol π UC-realizes a well-designed ideal functionality $\mathcal{F}$ if, for any PPT adversary $\mathcal{A}$, the ensembles $\mathsf{REAL}_{\pi,A,Z}$ and $\mathsf{IDEAL}_{F,S,Z}$ are indistinguishable.*

3. Security Model of SALRS in the UC Framework

In this section, we aim to define the security model of SALRS in the universal composability model by introducing the newly designed ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$. The definition of $\mathcal{F}_{\mathsf{PDPKS}}$ is presented in Figure 1.

We assume that this ideal functionality operates under a fixed system parameter, hence the Setup functionality interface. This omission eliminates the need for repetitive checks on the rationality of system parameters in subsequent interfaces.

Remark 3. *Our definition in the UC framework captures the correctness, soundness, and privacy of SALRS simultaneously. A formal proof establishing the existence of a UC-secure construction will be presented in Section 4.*

Linkable Ring Signature Supporting Stealth Address

The functionality $\mathcal{F}_{\mathsf{SALRS}}$ is parameterized with a fixed system parameter PP, and interacts with n participants $\mathsf{P}_1, \ldots, \mathsf{P}_n$ and a simulator S. The initialization of empty sets $L_{dpk,i}$ is performed for $i \in [n]$.

- **MasterKeyGen**: Upon receiving $(\mathsf{MasterKeyGen}, \mathsf{sid})$ from a party P_i:
  1. Send $(\mathsf{MasterKeyGen}, \mathsf{sid}, \mathsf{P}_i)$ to the simulator S. After S responses $(\mathsf{MasterKeyGened}, \mathsf{sid}, \mathsf{P}_i, \mathsf{mpk}_i)$, send $(\mathsf{MasterKeyGened}, \mathsf{sid}, \mathsf{mpk}_i)$.
  2. Record $(\mathsf{P}_i, \mathsf{mpk}_i)$.
- **DPKDerive**: Upon receiving $(\mathsf{DPKDerive}, \mathsf{sid}, \mathsf{mpk}'_i, \mathsf{P}_i)$ from a party P_j:
  1. If there is no record $(\mathsf{P}_i, \mathsf{mpk}_i)$ in memory such that $\mathsf{mpk}'_i = \mathsf{mpk}_i$, then ignore this request.
  2. Otherwise, send $(\mathsf{DPKDerive}, \mathsf{sid}, \mathsf{mpk}'_i, \mathsf{P}_i)$ to the simulator S. Upon receiving $(\mathsf{DPKDerived}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{P}_i)$ from the simulator S, send $(\mathsf{DPKDerived}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{P}_i)$ to P_j, and update $L_{\mathsf{dpk},i} = L_{\mathsf{dpk},i} \cup \{\mathsf{dpk}_i\}$.
- **DPKOwnerCheck**: Upon receiving $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk}_i)$ from a party P_i:
  1. Send $(\mathsf{DpkOwnercheck}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{P}_i)$ to the simulator S, and receive the response $(\mathsf{DPKOwnerChecked}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{P}_i, g)$ from S, where $g \in \{0, 1\}$.
  2. If P_i has not been compromised and $\mathsf{dpk}_i \in L_{\mathsf{dpk},i}$, set $f = 1$. Otherwise, set $f = g$. If $g = 1$, update $L_{\mathsf{dpk},i} = L_{\mathsf{dpk},i} \cup \{\mathsf{dpk}_i\}$. Otherwise, record $(\mathsf{dpk}_i, \mathsf{P}_i, 0)$.
  3. Send $(\mathsf{DPKOwnerChecked}, \mathsf{sid}, \mathsf{dpk}i, \mathsf{P}_i, f)$ to P_i.
- **DPKPublicCheck**: Upon receiving $(\mathsf{DPKPublicCheck}, \mathsf{sid}.\mathsf{dpk}_i)$ from a party P_j):
  1. Send $(\mathsf{DPKPublicCheck}, \mathsf{sid}, \mathsf{dpk}_i)$ to the simulator S, and receive a response $(\mathsf{DPKPublicChecked}, \mathsf{sid}, \mathsf{dpk}_i, g)$ from the simulator S, where $g \in \{0, 1\}$.
  2. If P_j has not been compromised and dpk_i belongs to some set $L_{\mathsf{dpk},i}$ (where $i \in [n]$), set $f = 1$. Otherwise, set $f = g$. If $g = 1$, update $L_{\mathsf{dpk},i} = L_{\mathsf{dpk},i} \cup \{\mathsf{dpk}_i\}$.
  3. Send $(\mathsf{DPKPublicChecked}, \mathsf{sid}, \mathsf{dpk}_i, f)$ to P_j.
- **Sign**: Upon receiving $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk}_i)$ from a party P_i:
  1. Send $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk}_i, \mathsf{P}_i)$ to the simulator S, and receive the response $(\mathsf{Signature}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk}_i)$ from S.
  2. If P_i is uncompromised, and either the d public key ring R is incorrectly formatted or $\mathsf{dpk}_i \notin L_{\mathsf{dpk},i}$, return an error message to P_i. Otherwise, check if there is a record $(M, R, \sigma, \mathsf{dpk}_i, 0)$ in memory. If found, output an error message to P_i. Otherwise, record the information $(M, R, \sigma, \mathsf{dpk}_i, 1)$ and return $(\mathsf{Signature}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk}_i)$ to P_i.
- **Verify**: Upon receiving $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$ from a party P_i:
  Send $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$ to the simulator S. Upon receiving $(\mathsf{Verified}, \mathsf{sid}, M, R, \sigma, f')$ from the simulator S, return $(\mathsf{Verified}, \mathsf{sid}, M, R, \sigma, f)$, where f is determined as follows:
  - If the derived public key ring R is well formed and there is information in memory $(M, R, \sigma, *, 1)$ where "$*$" serves as a wildcard, set $f = 1$.
  - Otherwise, if the derived public key ring R is well formed and there is no information about (M, R, σ) in memory, set $f = 0$.
  - Otherwise, if there is information $(M, R, \sigma, *, g)$, where "$*$" serves as a wildcard, set $f = g$.
  - Otherwise, set $f = f'$.
- **Link**: Upon receiving $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$ from a party P_i:
  Send $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$ to the simulator S. Upon receiving $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f')$ from the simulator S, return $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f)$, where f is determined as follows:
  If there is information in memory $(M_0, R_0, \sigma_0, *_1, 1)$ and $(M_1, R_1, \sigma_1, *_2, 1)$ where $*_1 = *_2$, set $f = 1$. Otherwise, set $f = f'$.

Figure 1. Ideal functionality of linkable ring signature supporting stealth addresses.

4. A UC-Secure SALRS Construction

In this section, we prove that the UC-security of SALRS defined above in Section 3 is equivalent to satisfying signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability simultaneously.

Let $\Sigma = (\mathsf{Setup}, \mathsf{MasterKeyGen}, \mathsf{DPKDerive}, \mathsf{DPKOwnerCheck}, \mathsf{DPKPublicCheck}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Link})$ denote the SALRS scheme. The protocol π_Σ is constructed from Σ, shown in Figure 2. Similar to the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$, it shares identical interfaces with the environment $\mathcal{Z}$.

Protocol π_Σ

- **Setup:** Upon receiving input $(\mathsf{Setup}, \mathsf{sid}, \mathsf{T})$, where T is a setup party, T acquires the system parameter PP by executing the **Setup** algorithm with a security parameter κ, and subsequently outputs PP.
- **MasterKeyGen:** Upon receiving input $(\mathsf{MasterKeyGen}, \mathsf{sid}, \mathsf{PP})$, a participant P_i executes the **MasterKeyGen** algorithm with the system parameters PP, generating a master key pair $(\mathsf{mpk}_i, \mathsf{msk}_i)$, and outputs the corresponding master public key mpk_i.
- **DPKDerive:** Upon receiving input $(\mathsf{DPKDerive}, \mathsf{sid}, \mathsf{mpk}_i)$, a participant P_j runs the **DPKDerive** algorithm with mpk_i and PP, generating a derived public key dpk_i corresponding to the master public key mpk_i.
- **DPKOwnerCheck:** Upon receiving input $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk}_i)$, a participant P_i executes the **DPKOwnerCheck** algorithm, determining a bit value $b \in \{0,1\}$, indicating whether dpk_i is derived from P_i's master public key mpk_i.
- **DPKPublicCheck:** Upon receiving input $(\mathsf{DPKPublicCheck}, \mathsf{sid}, \mathsf{dpk}_i)$ to a participant P_j, P_j executes the **DPKPublicCheck** algorithm to assess whether dpk_i is a well-formed derived public key derived from any master public key in the system, yielding a bit value $b \in \{0,1\}$.
- **Sign:** Upon receiving input $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk}_i)$ to a participant P_i, P_i executes the **Sign** algorithm, producing a signature σ.
- **Verify:** Upon receiving input $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$ to a participant P_j, P_j executes the **verification** algorithm, determining a bit value $b \in \{0,1\}$, where $b = 1$ denotes a valid signature, and $b = 0$ indicates an invalid one.
- **Link:** Upon receiving input $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$ to a participant P_j, P_j executes the **Link** algorithm, determining a bit value $b \in \{0,1\}$, where $b = 1$ signifies that the two sets of signatures are linkable, and $b = 0$ signifies the non-linkability.

Figure 2. An SALRS protocol π_Σ.

We establish equivalence by proving that a UC-secure SALRS scheme implies an SALRS scheme with signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability, and vice versa.

Lemma 1. *Let Σ be an SALRS scheme. If the corresponding protocol π_Σ securely realizes the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$, then the SALRS scheme Σ satisfies signer-linkability (SN-LINK), signer-non-slanderability (SN-NSL), signer-anonymity (SN-ANO), and master-public-key-unlinkability (MPK-UNL) simultaneously.*

Proof. We prove this lemma by contradiction. In other words, if Σ lacks signer-linkability, signer-non-slanderability, signer-anonymity, or master-public-key-unlinkability, then π_Σ cannot UC-realize the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$.

Firstly, if Σ lacks signer-linkability, there exists an adversary $\mathcal{G}$ that can break the signer-linkability property of Σ with a non-negligible advantage. In other words, there exists a PPT adversary $\mathcal{A}$, for any ideal world simulator S, and an environment Z that, with the assistance of $\mathcal{G}$, can distinguish $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$ with a non-negligible probability. The process of the environment $\mathcal{Z}$ is as follows:

1. $\mathcal{Z}$ activates the Setup Party T with information $(\mathsf{Setup}, \mathsf{sid}, \mathsf{T})$, obtaining system parameters PP, and sends PP to adversary $\mathcal{G}$.
2. $\mathcal{Z}$ receives k (where $k \geq 2$) tuples $(M_i^*, R_i^*, \sigma_i^*)$ $(i \in [k])$ from adversary $\mathcal{G}$, consisting of messages, well-formed derived public key rings, and signatures.

In step 2, because adversary $\mathcal{G}$ can break the signer-linkability of Σ, the k tuples received by $\mathcal{Z}$ satisfy the following conditions:

1. $\mathsf{Verify}(M_i^*, R_i^*, \sigma_i^*) = 1$, where $i \in [k]$;
2. $\forall i, j \in [k] s.t. i \neq j, \mathsf{Link}(M_i^*, R_i^*, \sigma_i^*, M_j^*, R_j^*, \sigma_j^*) = 0$;
3. $\left| \cup_{i \in [k]} R_i^* \right| \leq k$.

When $\mathcal{Z}$ executes in the real world, all these conditions can be verified. However, when $\mathcal{Z}$ executes in the ideal world, since the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$ does not store relevant information, the first condition cannot be verified. Therefore, $\mathcal{Z}$ distinguishes between the real and ideal worlds, and the probability that $\mathcal{Z}$ distinguishes between the real and ideal worlds is equal to the probability that $\mathcal{G}$ can break the signer-linkability. Hence, if Σ does not satisfy signer-linkability, then π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$.

Secondly, if Σ lacks signer-non-slanderability, there exists an adversary $\mathcal{G}$ that can break the signer-non-slanderability property of Σ with a non-negligible advantage. In other words, there exists a PPT adversary $\mathcal{A}$, for any ideal-world simulator S, and an environment $\mathcal{Z}$ that, with the assistance of G, can distinguish $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$ with a non-negligible probability. The interaction process of the environment $\mathcal{Z}$ is as follows:

1. $\mathcal{Z}$ activates the setup party T with information $(\mathsf{Setup}, \mathsf{sid}, \mathsf{T})$, obtaining system parameters PP, and sends PP to adversary $\mathcal{G}$.
2. When $\mathcal{Z}$ receives a query on the master public key of a participant P_i from adversary $\mathcal{G}$, $\mathcal{Z}$ activates participant P_i to obtain its master public key and sends it to $\mathcal{G}$. $\mathcal{G}$ can inquire about the master public key of any participant.
3. When $\mathcal{Z}$ receives a query from adversary $\mathcal{G}$ regarding whether a given derived public key dpk_i is derived from a given master public key $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk}_i, P_i)$, $\mathcal{Z}$ activates participant P_i to obtain the check result and sends it to $\mathcal{G}$.
4. When $\mathcal{Z}$ receives a signature query about (M, R, dpk_i) from adversary $\mathcal{G}$, $\mathcal{Z}$ activates the owner of the derived public key dpk_i to obtain the signature result and sends it to $\mathcal{G}$.
5. When $\mathcal{Z}$ receives two well-formed tuples $(\hat{M}, \hat{R}, \hat{\sigma})$ and (M^*, R^*, σ^*) from adversary $\mathcal{G}$, where (1) (M^*, R^*, σ^*) can be verified by signature, (2) $(\hat{M}, \hat{R}, \hat{\sigma})$ is the signature result of $\mathcal{G}$'s query to $\mathcal{Z}$ about a derived public key $\hat{\mathsf{dpk}}$, (3) $(\hat{M}, \hat{R}, \hat{\sigma})$ is not the signature result of $\mathcal{G}$'s query to $\mathcal{Z}$ about derived public key $\hat{\mathsf{dpk}}$, and (4) these two tuples can pass the linkable verification, $\mathcal{Z}$ outputs 0 and halts. Otherwise, $\mathcal{Z}$ activates the party to return the linkable verification bit. $\mathcal{Z}$ obtains such tuples, and if $\mathcal{Z}$ is interacting with $\mathcal{A}$ and π_Σ in the real world, $\mathcal{Z}$ will output 1, since signature verification and linkable verification are valid. If $\mathcal{Z}$ is interacting with S and $\mathcal{F}_{\mathsf{SALRS}}$ in the ideal world, $\mathcal{Z}$ will output 0 because the ideal function $\mathcal{F}_{\mathsf{SALRS}}$ does not record $(M^*, R^*, *, \sigma^*)$, so signature verification cannot pass, or $\mathcal{F}_{\mathsf{SALRS}}$ records $(M^*, R^*, *, \sigma^*)$, but $* \neq \hat{\mathsf{dpk}}$, so linkable verification cannot pass.

Since $\mathcal{G}$ can break the signer-non-slanderability property of Σ with a non-negligible probability, the probability that $\mathcal{Z}$ outputs 1 when interacting with the real model is also non-negligible. Therefore, $\mathcal{Z}$ can distinguish the interaction with the real model and the ideal model with a non-negligible probability. In other words, if Σ lacks signer-non-slanderability, then π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$.

Thirdly, if Σ lacks signer-anonymity, there exists an adversary $\mathcal{G}$ that can break the signer-anonymous property of Σ with a non-negligible advantage. In other words, there exists a PPT adversary $\mathcal{A}$, for any ideal-world simulator S, and an environment $\mathcal{Z}$ that, with the assistance of $\mathcal{G}$, can distinguish $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$ with a non-negligible probability. The interaction process of the environment $\mathcal{Z}$ is as follows:

1. Activate parties $\{\mathsf{P}_i\}_{i \in [poly(\kappa)]}$ with the message $(\mathsf{Masterkeygen}, \mathsf{sid}, \mathsf{PP})$ to obtain individual master public keys $\{\mathsf{mpk}_i\}_{i \in [poly(\kappa)]}$.
2. Send $\{\mathsf{mpk}_i\}_{i \in [poly(\kappa)]}$ to $\mathcal{G}$, and play the roles of oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot, \cdot)$ for adding derived public keys and the signing oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot, \cdot)$. Initialize the empty set $L_{\mathsf{dpk}} = \emptyset$.
3. Receive a message M^*, a well-formed derived public key ring R^*, and two derived public keys dpk_{i_0} and dpk_{i_1} from $\mathcal{G}$, satisfying the following: (1) $\mathsf{dpk}_{i_0}, \mathsf{dpk}_{i_1} \in R^* \cap L_{\mathsf{dpk}}$, and (2) neither dpk_{i_0} or dpk_{i_1} is queried before as an input by oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot, \cdot)$.
4. Randomly choose a bit $b \in \{0, 1\}$, run the $\mathsf{DPKOwnerCheck}$ algorithm to obtain the participant corresponding to the selected target derived public key dpk_{i_b}, and activate this participant to obtain a signature $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_{i_b}, \mathsf{mpk}, \mathsf{msk})$, where $(\mathsf{mpk}, \mathsf{msk})$ is the master key pair corresponding to dpk_{i_b}. Send this signature σ to $\mathcal{G}$.
5. Continue to play the roles of oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot, \cdot)$ and oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot, \cdot, \cdot)$ for the adversary $\mathcal{G}$.

6. Receive b' from $\mathcal{G}$, output 1 if $b \neq b'$, otherwise output 0 and halt.

In step 2, adversary $\mathcal{G}$ initiates queries $q_1, \ldots, q_m$, where query q_l is one of the following:

- Oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot,\cdot)$: $\mathcal{Z}$ receives a derived public key adding request concerning dpk and the master public key mpk_i. $\mathcal{Z}$ sends a derived public key owner check request regarding this information to the participant P_i corresponding to the master public key mpk_i, obtaining the return value $b \leftarrow \mathsf{DPKOwnerCheck}(\mathsf{dpk}, \mathsf{mpk}_i, \mathsf{msk}_i)$. If $b = 1$, update $L_{\mathsf{dpk}} = L\mathsf{dpk} \cup \{\mathsf{dpk}\}$. Return the result b to $\mathcal{G}$.
- Oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$: $\mathcal{Z}$ receives a signature request concerning the message M, a well-formed derived public key ring R, and a derived public key $\mathsf{dpk} \in R \cap L_{\mathsf{dpk}}$. $\mathcal{Z}$ queries the owner of the derived public key dpk and activates the owner of the derived public key dpk with this signature request. $\mathcal{Z}$ receives the returned signature information $\sigma \leftarrow \mathsf{Sign}(M, R, \mathsf{dpk}, \mathsf{mpk}_i, \mathsf{msk}_i)$, where $(\mathsf{mpk}_i, \mathsf{msk}_i)$ is the master public–private key pair corresponding to dpk. Return the signature σ to $\mathcal{G}$.

These query requests may be adaptive, meaning that each query q_l may be determined based on the answers to previous queries $q_1, \ldots, q_{l-1}$.

In step 5, adversary $\mathcal{G}$ initiates more queries $q_{m+1}, \ldots, q_n$, where q_l may be adaptively chosen as in step 2, except that $\mathcal{O}^{\mathsf{Sign}}(\mathsf{dpk}_{i_0}, \cdot, \cdot)$ and $\mathcal{O}^{\mathsf{Sign}}(\mathsf{dpk}_{i_1}, \cdot, \cdot)$ cannot be queried.

When $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ, $\mathcal{Z}$ in step 4 obtains a signature $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_{i_b}, \mathsf{mpk}, \mathsf{msk})$, and $\mathcal{G}$ can break the signer-anonymity with a non-negligible advantage. When $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ, we use $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}]$ to denote the probability that $\mathcal{Z}$ outputs 1.

$$\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}] \\
=&\frac{1}{2}\big(1 - \Pr\big[b' = 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_{i_0}, \mathsf{mpk}, \mathsf{msk})\big]\big) \\
&+ \frac{1}{2}\Pr\big[b' = 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_{i_1}, \mathsf{mpk}, \mathsf{msk})\big] \\
=&\frac{1}{2} + \frac{1}{2}\Big(\Pr\Big[\mathbf{Exp}^{\mathsf{snano},1}_{\Sigma,\mathcal{G}}(\kappa) = 1\Big] - \Pr\Big[\mathbf{Exp}^{\mathsf{snano},0}_{\Sigma,\mathcal{G}}(\kappa) = 1\Big]\Big) > \frac{1}{2} + \frac{1}{2}\mathsf{negl}(\kappa).
\end{aligned}$$

In contrast, when $\mathcal{Z}$ interacts with the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$ and any adversary, the instance of $\mathcal{G}$'s perspective is statistically independent of b. In this case, the probability that $b = b'$ is exactly one-half. $\mathcal{G}$'s perspective is independent of b; it includes all derived public-key-checking algorithms and signing algorithms. The σ randomly generated by S is independent of b, and the oracle queries provided by $\mathcal{Z}$ are also independent of b.

When $\mathcal{Z}$ interacts with S and the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$, we denote by $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}]$ the probability that $\mathcal{Z}$ outputs 1.

$$\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}] \\
=&\frac{1}{2}\big(1 - \Pr\big[b' = 1 | \sigma \leftarrow \mathsf{S}\big]\big) + \frac{1}{2}\Pr\big[b' = 1 | \sigma \leftarrow \mathsf{S}\big] \\
=&\frac{1}{2}\big(1 - \Pr\big[b' = 1 | \sigma \leftarrow \mathsf{S}\big] + \Pr\big[b' = 1 | \sigma \leftarrow \mathsf{S}\big]\big) = \frac{1}{2}.
\end{aligned}$$

Therefore, the probability $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}] - [\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}] > \frac{1}{2}\mathsf{negl}(\kappa)$. Thus, $\mathcal{Z}$ can distinguish $(\pi_\Sigma, \mathcal{A})$ and $(\mathcal{F}_{\mathsf{SALRS}}, \mathsf{S})$ with a non-negligible probability, proving that UC-secure SALRS implies signer-anonymity of SALRS.

Fourthly, if Σ lacks master-public-key-unlinkability, there exists an adversary $\mathcal{G}$ that can break the master-public-key-unlinkability property of Σ with a non-negligible advantage. In other words, there exists a PPT adversary $\mathcal{A}$, for any ideal world simulator $\mathcal{S}$, and an environment $\mathcal{Z}$ that, with the assistance of $\mathcal{G}$, can distinguish $(\mathcal{F}_{\mathsf{SALRS}}, \mathsf{S})$ and $(\pi_\Sigma, \mathcal{A})$ with a non-negligible probability. The interaction process of the environment $\mathcal{Z}$ is as follows:

1. Activate each participant $\{P_i\}_{i\in[poly(\kappa)]}$ with the message ($\mathsf{Masterkeygen}$, sid, PP), obtaining the master public keys $\{\mathsf{mpk}_i\}_{i\in[poly(\kappa)]}$ for each participant, and send them to $\mathcal{G}$.
2. Play the roles of the oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot,\cdot)$ and a signature oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ for adversary $\mathcal{G}$ during the interaction. Initialize an empty set $L_{\mathsf{dpk}} = \emptyset$.
3. $\mathcal{G}$ sends two master public keys mpk_{i_0} and mpk_{i_1} to $\mathcal{Z}$. $\mathcal{Z}$ randomly chooses a bit $b \leftarrow \{0,1\}$, selects an arbitrary participant P_r, and activates P_r with ($\mathsf{DPKDerive}$, sid, mpk_{i_b}), obtaining $\mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_b})$.
4. Send dpk^* to $\mathcal{G}$ as the target derived public key.
5. Continue playing the role of an oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot,\cdot)$ and a signature oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ for adversary $\mathcal{G}$ during the interaction, except that queries $\mathcal{O}^{\mathsf{DPKAdd}}(\mathsf{dpk}^*, \mathsf{mpk}_{i_j})$ where $j \in \{0,1\}$ cannot be made.
6. $\mathcal{G}$ outputs b' as the guess result. If $b = b'$, output 1; otherwise, output 0 and halt.

In step 2, adversary $\mathcal{G}$ initiates queries $q_1, \ldots, q_m$, where query q_l can be one of the following:

- Oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot,\cdot)$: When $\mathcal{Z}$ receives a query from $\mathcal{G}$ about whether a given derived public key dpk belongs to a certain master public key mpk_i, $\mathcal{Z}$ sends this information to the participant P_i corresponding to mpk_i. When $\mathcal{Z}$ receives the result $b \leftarrow \mathsf{DPKOwnerCheck}(\mathsf{dpk}, \mathsf{mpk}_i, \mathsf{msk}_i)$ from participant P_i, if $b = 1$, update $L_{\mathsf{dpk}} = L\mathsf{dpk} \cup \{\mathsf{dpk}\}$. Submit the result b to $\mathcal{G}$.

These query requests may be adaptive, meaning that each query q_l may depend on the responses to previous queries $q_1, \ldots, q_{l-1}$.

In step 5, adversary $\mathcal{G}$ initiates additional queries $q_{m+1}, \ldots, q_n$, where q_l may be adaptively chosen like in step 2, except for queries $\mathcal{O}^{\mathsf{DPKAdd}}(dpk^*, \mathsf{dpk}_{i_j})$, where $j \in \{0,1\}$, cannot be made.

When $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ, in step 3, $\mathcal{Z}$ obtains $\mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_b})$. $\mathcal{G}$ can break the master-public-key-unlinkability with a non-negligible advantage. When $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ, we use $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}]$ to denote the probability that $\mathcal{Z}$ outputs 1.

$$
\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}] \\
=&\Pr\big[\mathsf{mpk}_{i_b} = \mathsf{mpk}_{i_0}\big] \Pr\big[b' = 0 | \mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_0})\big] \\
&+ \Pr\big[\mathsf{mpk}_{i_b} = \mathsf{mpk}_{i_1}\big] \Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_1})\big] \\
=&\frac{1}{2}\big(1 - \Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_0})\big]\big) + \frac{1}{2}\Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_{i_1})\big] \\
=&\frac{1}{2} + \frac{1}{2}\Big(\Pr\Big[\mathbf{Exp}^{mpkunl,0}_{\Sigma,\mathcal{G}}(\kappa) = 1\Big] - \Pr\Big[\mathbf{Exp}^{mpkunl,1}_{\Sigma,\mathcal{G}}(\kappa) = 1\Big]\Big) > \frac{1}{2} + \frac{1}{2}\mathsf{negl}(\kappa).
\end{aligned}
$$

In contrast, when $\mathcal{Z}$ interacts with the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$ and any adversary, the perspective of the instance $\mathcal{G}$ is statistically independent of b. In this case, the probability that $b = b'$ is exactly one-half. The derived public key dpk^* generated by S is independent of b, and the queries provided by $\mathcal{Z}$ are also independent of b. When $\mathcal{Z}$ interacts with S and $\mathcal{F}_{\mathsf{SALRS}}$ in the ideal world, let $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}]$ denote the probability that $\mathcal{Z}$ outputs 1.

$$
\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}] \\
=&\Pr\big[\mathsf{mpk}_{i_b} = \mathsf{mpk}_{i_0}\big] \Pr\big[b' = 0 | \mathsf{dpk}^* \leftarrow \mathsf{S}\big] + \Pr\big[\mathsf{mpk}_{i_b} = \mathsf{mpk}_{i_1}\big] \Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{S}\big] \\
=&\frac{1}{2}\big(1 - \Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{S}\big]\big) + \frac{1}{2}\Pr\big[b' = 1 | \mathsf{dpk}^* \leftarrow \mathsf{S}\big] = \frac{1}{2}.
\end{aligned}
$$

Therefore, $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{REAL}] - \Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \mathsf{IDEAL}] > \frac{1}{2}\mathsf{negl}(\kappa)$. Thus, $\mathcal{Z}$ can distinguish $(\pi_\Sigma, \mathcal{A})$ from $(\mathcal{F}_{\mathsf{SALRS}}, \mathsf{S})$ with a non-negligible probability, demonstrating that UC-secure SALRS inherently implies the non-linkability of public keys in SALRS.

In conclusion, if π_Σ UC-realizes ($\mathcal{F}_{\mathsf{SALRS}}$, then Σ satisfies the properties of signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability. □

Lemma 2. *If an SALRS scheme Σ satisfies signer-linkability, signer-non-slanderability, signer-anonymity, an d master-public-key-unlinkability simultaneously, the corresponding protocol π_Σ securely realizes the ideal functionality $\mathcal{F}_{SALRS}$.*

Proof. We establish the proof through a method of contradiction. In other words, if π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$, then Σ fails to satisfy at least one of the properties: signer-linkability, signer-non-slanderability, signer-anonymity, or master-public-key-unlinkability.

Firstly, we claim that if π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$, while satisfying the other three properties, it can be deduced that Σ does not satisfy signer-linkability. In more detail, we assume the existence of an adversary $\mathcal{A}$ in the real world such that for any ideal world adversary S, there exists an environment $\mathcal{Z}$ capable of distinguishing $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$. If this holds true, then there exists an adversary $\mathcal{B}$ that simulates the simulator S and the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$, using the environment $\mathcal{Z}$ to distinguish between the ideal and real world.

$\mathcal{B}$ simulates the ideal adversary S in the following manner: Firstly, $\mathcal{B}$ obtains the public key mpk_i of participant P_i from $\mathcal{Z}$.

1. Upon receiving input from the environment $\mathcal{Z}$, $\mathcal{B}$ forwards this input to $\mathcal{A}$ and replicates $\mathcal{A}$'s output as its own output.
2. Upon receiving $(\mathsf{DPKDerive}, \mathsf{sid}, \mathsf{mpk}_i')$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ first checks if $\mathsf{mpk}_i' = \mathsf{mpk}_i$. If not, it ignores this information; otherwise, it runs the algorithm $\mathsf{DPKDerive}(\mathsf{mpk}_i)$ to obtain a derived public key dpk_i corresponding to mpk_i.
3. Upon receiving $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk}_i)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ queries the derived public key adding oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot)$ to verify whether dpk_i is derived from mpk_i and returns the verification result $(\mathsf{DPKOwnerChecked}, \mathsf{sid}, \mathsf{dpk}_i, f)$.
4. Upon receiving $(\mathsf{DPKPublicCheck}, \mathsf{sid}, \mathsf{dpk}_i)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ runs the corresponding algorithm and returns the verification result.
5. Upon receiving $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk}_i)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ queries the signature oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ to obtain a signature σ for the message M, the ring R, and the derived public key dpk_i, and returns $(\mathsf{Signature}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk}_i)$.
6. Upon receiving $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ runs the verification algorithm to obtain a verification value f and returns $(\mathsf{Verified}, \mathsf{sid}, M, R, \sigma, f)$.
7. Upon receiving $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ runs the corresponding linking verification algorithm to obtain a verification value f and returns $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f)$.

Clearly, in the above interaction, through querying oracles and invoking algorithms, the simulated S and $\mathcal{F}_{\mathsf{SALRS}}$ by $\mathcal{B}$ are indistinguishable from the real S and $\mathcal{F}_{\mathsf{SALRS}}$.

When the environment $\mathcal{Z}$ activates the participant P_j with $(\mathsf{Link}, \mathsf{sid}, M_0^*, R_0^*, \sigma_0^*, M_1^*, R_1^*, \sigma_1^*)$, $\mathcal{B}$ verifies whether this information is linkable. If the linkability verification fails, and at the same time, $\mathcal{B}$ can successfully verify the signatures for the tuples $(M_0^*, R_0^*, \sigma_0^*)$ and $(M_1^*, R_1^*, \sigma_1^*)$ while having queried the signature oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ about $(M_0^*, R_0^*, *_0^*)$ and $(M_1^*, R_1^*, *_1^*)$, obtaining signatures σ_0^* and σ_1^*, where $*_0 \neq *_1$, then $\mathcal{B}$ outputs $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$ and halts. In other words, $\mathcal{B}$ has obtained a set of information that breaks the linkability of signers. Otherwise, $\mathcal{B}$ continues the simulation.

If $\mathcal{B}$ can obtain such a set of information, then for the input $(\mathsf{Link}, \mathsf{sid}, M_0^*, R_0^*, \sigma_0^*, M_1^*, R_1^*, \sigma_1^*)$, if $\mathcal{Z}$ interacts with the real-world protocol π_Σ, the observed output by $\mathcal{Z}$ is 1; if $\mathcal{Z}$ executes in the ideal world, $\mathcal{Z}$ observes an output of 0. In other words, $\mathcal{Z}$ can distinguish whether it is interacting with the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$ or the implemented protocol π_Σ. Therefore, if the probability of $\mathcal{B}$ successfully breaking the signer-linkability is negligible, then the probability that the environment $\mathcal{Z}$ can distinguish between the real world and the ideal world is also negligible, contradicting the assumption.

Secondly, we claim that if π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$, while satisfying the other three properties, it can be deduced that Σ does not satisfy signer-non-slanderability. In more detail, we assume the existence of an adversary $\mathcal{A}$ in the real world such that for any simulator S, there exists an environment $\mathcal{Z}$ capable of distinguishing $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ from $(\mathcal{A}, \pi_\Sigma)$. This assumption leads to the existence of an adversary $\mathcal{B}$ that simulates the ideal world simulator S and ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$, attempting to distinguish between the ideal and the real world by interacting with the environment $\mathcal{Z}$.

$\mathcal{B}$ simulates the ideal adversary S in the following manner: Firstly, $\mathcal{B}$ obtains the public key $\{\mathsf{mpk}_i\}_{i\in[poly(\kappa)]}$ of all participants from $\mathcal{Z}$.

1. Upon receiving input from the environment $\mathcal{Z}$, $\mathcal{B}$ forwards this input to $\mathcal{A}$ and replicates $\mathcal{A}$'s output as its own output.
2. Upon receiving $(\mathsf{DPKDerive}, \mathsf{sid}, \mathsf{mpk}'_i)$ from $\mathcal{F}_{\mathsf{SALRS}}$, $\mathcal{B}$ first checks if $\mathsf{mpk}'_i = \mathsf{mpk}_i$. If not, it ignores the message; otherwise, it runs the algorithm $\mathsf{DPKDerive}(\mathsf{mpk}_i)$ to obtain a derived public key dpk_i corresponding to mpk_i.
3. Upon receiving $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{mpk}_i)$, $\mathcal{B}$ queries the oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot)$ to verify whether dpk_i is derived from mpk_i and returns the verification result $(\mathsf{DPKOwnerChecked}, \mathsf{sid}, \mathsf{dpk}_i, \mathsf{mpk}_i, f)$.
4. Upon receiving $(\mathsf{DPKPublicCheck}, \mathsf{sid}, \mathsf{dpk}_i)$, $\mathcal{B}$ runs the corresponding algorithm and returns the verification result.
5. Upon receiving $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk}_i)$, $\mathcal{B}$ queries the signing oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ to obtain the signature σ for the message M, context R, and derived public key dpk_i, and returns $(\mathsf{Signature}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk}_i)$.
6. Upon receiving $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$, $\mathcal{B}$ runs the signature verification algorithm to obtain the verification value f and returns $(\mathsf{Verified}, \mathsf{sid}, M, R, \sigma, f)$.
7. Upon receiving $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$, $\mathcal{B}$ runs the corresponding linkability verification algorithm to obtain the verification value f and returns $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f)$.

Clearly, in the above interaction, through querying oracles and invoking algorithms, the simulations of S and $\mathcal{F}_{\mathsf{SALRS}}$ by $\mathcal{B}$ are indistinguishable from the actual S and $\mathcal{F}_{\mathsf{SALRS}}$.

When the environment $\mathcal{Z}$ outputs two tuples $(\hat{M}, \hat{R}, \hat{\sigma})$ and (M^*, R^*, σ^*), these tuples satisfy the following conditions: (1) (M^*, R^*, σ^*) can be verified by signature verification; (2) $(\hat{M}, \hat{R}, \hat{\sigma})$ is the signature result queried by $\mathcal{B}$ regarding a certain derived public key $\hat{\mathsf{dpk}}$; (3) (M^*, R^*, σ^*) is not the signature result queried by $\mathcal{B}$ regarding the derived public key $\hat{\mathsf{dpk}}$; (4) these two tuples can be successfully verified by the linkability verification. In this case, $\mathcal{B}$ outputs this message pair and halts, indicating that $\mathcal{B}$ has obtained a pair of messages that can defame the signer. Otherwise, $\mathcal{B}$ continues the simulation.

If $\mathcal{B}$ can obtain such a pair of messages that can defame the signer, then if $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ in the real world, the outputs observed by $\mathcal{Z}$ are 1 due to the effectiveness of signature verification and linkability verification. If $\mathcal{Z}$ interacts with S and $\mathcal{F}_{\mathsf{SALRS}}$ in the ideal world, signature verification cannot pass, since $(M^*, R^*, *, \sigma^*)$ is not recorded in the ideal functionality $\mathcal{F}_{\mathsf{SALRS}}$. Therefore, $\mathcal{Z}$ observes an output of 0. Alternatively, if $\mathcal{F}_{\mathsf{SALRS}}$ records $(M^*, R^*, *, \sigma^*)$, but $* \neq \hat{\mathsf{dpk}}$?, linkability verification cannot pass, and $\mathcal{Z}$ observes an output of 0. In this way, $\mathcal{Z}$ can distinguish whether it is interacting in the real or ideal world. Therefore, if the probability that $\mathcal{B}$ can slander the signer is negligible, then the probability that $\mathcal{Z}$ can distinguish between the real and ideal worlds is also negligible, contradicting the assumption.

Thirdly, we claim that if π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$, while satisfying the other three properties, it can be deduced that Σ does not satisfy signer-anonymity. In other words, there exists an adversary $\mathcal{B}$, assisted by an environment $\mathcal{Z}$, capable of breaking the signer-anonymity property of Σ. To elaborate further, we assume the existence of an adversary $\mathcal{A}$ in the real world such that for any adversary S in the ideal world, there exists an environment $\mathcal{Z}$ capable of distinguishing $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ from $(\mathcal{A}, \pi_\Sigma)$ for any fixed security parameter κ and fixed input z:

$$\left|\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{SALRS}},\mathsf{S},\mathcal{Z}}(\kappa, z) - \mathsf{REAL}_{\pi_\Sigma,\mathcal{A},\mathcal{Z}}(\kappa, z)\right| > \mathsf{negl}(\kappa). \tag{3}$$

We demonstrate that the adversary $\mathcal{G}_h$ possesses an advantage in the signer-anonymity game, denoted as $\mathbf{Adv}^{\mathsf{snano}}_{\Sigma,\mathcal{G}_h}(\kappa) > \frac{\mathsf{negl}(\kappa)}{l}$, where l is the total number of signed messages. The public keys of the participants $\{\mathsf{mpk}_i\}_{i\in[poly(\kappa)]}$ are sent to $\mathcal{G}_h$ and $\mathcal{Z}$, allowing $\mathcal{G}_h$ to make queries to the two mentioned oracles. $\mathcal{G}_h$ conveys a message M^*, and a correctly formatted derived public key ring R^* to $\mathcal{Z}$. $\mathcal{G}_h$ simulates the operation of the environment $\mathcal{Z}$ similarly to the system running $\pi_\Sigma / \mathcal{F}_{\mathsf{SALRS}}$ as follows.

1. Whenever participant P_j is activated with input $(\mathsf{DPKDerive}, \mathsf{sid})$, $\mathcal{G}_h$ instructs P_j to return the corresponding derived public key. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
2. Whenever participant P_i is activated with input $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk})$, $\mathcal{G}_h$ instructs P_i to return the corresponding check result. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
3. Whenever participant P_j is activated with input $(\mathsf{DPKPublicCheck}, \mathsf{sid}, \mathsf{dpk})$, $\mathcal{G}_h$ instructs P_j to return the corresponding check result. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
4. For the first $h-1$ instances, $\mathcal{Z}$ requests signatures on $M^*, R^*, \text{and } \mathsf{dpk}_n$, where $n \in [h-1]$. $\mathcal{G}_h$ instructs the signing party to return a signature $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_n, \mathsf{mpk}, \mathsf{msk})$, where $(\mathsf{mpk}, \mathsf{msk})$ is the public–private key pair corresponding to dpk_n.
5. For the h-th instance, $\mathcal{Z}$ requests a signature on $M^*, R^*, \text{and } \mathsf{dpk}_h$. $\mathcal{G}_h$ randomly selects an honestly derived public key dpk_r from the set R^* and queries the oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ with information $(M^*, R^*, \mathsf{dpk}_h, \mathsf{dpk}_r)$ to obtain a signature σ in return. That is, during execution, when $b = 0$, $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_h, \mathsf{mpk}_h, \mathsf{msk}_h)$; when $b = 1$, $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_r, \mathsf{mpk}_r, \mathsf{msk}_r)$.
6. For the remaining $l-h$ instances, $\mathcal{Z}$ requests signatures on $M^*, R^*, \text{and } \mathsf{dpk}_n$, where $n \in [l]\backslash[h]$. $\mathcal{G}_h$ instructs the signing party P_r to return a signature $\sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_r, \mathsf{mpk}_r, \mathsf{msk}_r)$, where $(\mathsf{mpk}_r, \mathsf{msk}_r)$ is the master public–private key pair corresponding to dpk_r, and P_r is the owner of dpk_r.
7. Whenever participant P_j is activated with input $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$, $\mathcal{G}_h$ instructs P_j to output the execution result $(\mathsf{sid}, M, R, \sigma, f)$ to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
8. Whenever participant P_j is activated with input $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$, $\mathcal{G}_h$ instructs P_j to output the execution result $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f)$ to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
9. When $\mathcal{Z}$ halts, $\mathcal{G}_h$ outputs the output value of $\mathcal{Z}$ and halts.

We analyze the success probability of $\mathcal{G}_h$ using the methodology of hybrid argument. For $j \in \{0, \dots, l\}$, let $\mathbf{Env}_j$ represent the event: $\mathcal{Z}$ interacts with S in the ideal process, except that the first j signatures are generated by the truly derived public key dpk_i rather than an arbitrarily chosen derived public key $\mathsf{dpk}r$. Let H_j be $\Pr[\mathcal{Z} \to 1 | \mathbf{Env}_j]$.

We easily observe that H_0 is equivalent to the probability of $\mathcal{Z}$ outputting 1 in the ideal world, and H_l is equivalent to the probability of $\mathcal{Z}$ outputting 1 in the real world. Moreover, during the execution of $\mathcal{G}_h$, if $\mathcal{G}_h$ obtains a σ value from its signing oracle that is generated by the actual derived public key dpk_i, the probability of $\mathcal{Z}$ outputting 1 is equivalent to H_h. If σ is generated from an arbitrarily chosen honest derived public key dpk_r, the probability of $\mathcal{Z}$ outputting 1 is equivalent to H_{h-1}. The detailed process is as follows:

$$\begin{aligned} H_0 &= \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{SALRS}},\mathsf{S},\mathcal{Z}}(\kappa, z) \\ H_l &= \mathsf{REAL}_{\pi_\Sigma,\mathcal{A},\mathcal{Z}}(\kappa, z) \\ H_{h-1} &= \Pr[\mathcal{G}_h \to 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_r, \mathsf{mpk}_r, \mathsf{msk}_r)] \\ H_h &= \Pr[\mathcal{G}_h \to 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_i, \mathsf{mpk}_i, \mathsf{msk}_i)]. \end{aligned} \tag{4}$$

$$\begin{aligned} \sum_{i=1}^{l} |H_{i-1} - H_i| \geq \left| \sum_{i=1}^{l} (H_{i-1} - H_i) \right| &= |H_0 - H_l| \\ &= \left| \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{SALRS}},\mathsf{S},\mathcal{Z}}(\kappa, z) - \mathsf{REAL}_{\pi_\Sigma,\mathcal{A},\mathcal{Z}}(\kappa, z) \right| > \mathsf{negl}(\kappa). \end{aligned} \tag{5}$$

Therefore, there exists some $h \in \{0, \ldots, l\}$ such that $|H_{h-1} - H_h| > \frac{\mathsf{negl}(\kappa)}{l}$. Here, without loss of generality, we assume $H_{h-1} - H_h > \frac{\mathsf{negl}(\kappa)}{l}$. Thus, the advantage of the adversary $\mathcal{G}_h$ is as follows:

$$\begin{aligned} \mathbf{Adv}^{\mathsf{snano}}_{\Sigma,\mathcal{G}_h}(\kappa) =& \Pr\left[\mathbf{Exp}^{\mathsf{snano},1}_{\Sigma,\mathcal{A}}(\kappa) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{snano},0}_{\Sigma,\mathcal{A}}(\kappa) = 1\right] \\ =& \Pr[\mathcal{G}_h \to 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_i, \mathsf{mpk}_i, \mathsf{msk}_i)] \\ &- \Pr[\mathcal{G}_h \to 1 | \sigma \leftarrow \mathsf{Sign}(M^*, R^*, \mathsf{dpk}_r, \mathsf{mpk}_r, \mathsf{msk}_r)] \\ =& H_h - H_{h-1} > \frac{\mathsf{negl}(\kappa)}{l} \end{aligned} \tag{6}$$

This implies that $\mathcal{G}$ has a non-negligible advantage with respect to κ, as l is polynomially bounded in κ. Therefore, if the environment $\mathcal{Z}$ can distinguish between the real and ideal worlds, there exists an adversary $\mathcal{B}$ that, under the help of the environment $\mathcal{Z}$, breaks the signer-anonymity of Σ.

Finally, we claim that if π_Σ cannot UC-realize $\mathcal{F}_{\mathsf{SALRS}}$ while satisfying the other three properties, it can be deduced that Σ does not satisfy master-public-key-unlinkability. More specifically, we assume the existence of an adversary $\mathcal{A}$ in the real world such that, for any ideal-world adversary S, there exists an environmental machine $\mathcal{Z}$, which can distinguish $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ from $(\mathcal{A}, \pi_\Sigma)$ for any fixed security parameter κ and fixed input z, as shown in Equation (3).

We demonstrate that the adversary $\mathcal{G}_h$ exhibits an advantage in the game of master-public-key-unlinkability, denoted as $\mathbf{Adv}^{\mathsf{mpkunl}}_{\Sigma,\mathcal{G}_h}(\kappa) > \mathsf{negl}(\kappa)/l$, where l is the total number of generated target derived public keys. The public keys of participants, denoted as $\{\mathsf{mpk}_i\}_{i \in [poly(\kappa)]}$, are sent to both $\mathcal{G}_h$ and $\mathcal{Z}$, allowing $\mathcal{G}_h$ to make queries to the aforementioned two oracles. $\mathcal{G}_h$ simulates the environment $\mathcal{Z}$ in a manner analogous to the execution of $\pi^\Sigma / \mathcal{F}_{\mathsf{SALRS}}$.

1. For the first $h-1$ queries, $\mathcal{Z}$ requests participant P_j to provide a derived public key dpk_n related to mpk_i, where $n \in [h-1]$. $\mathcal{G}_h$ instructs P_j to execute the corresponding algorithm and return $\mathsf{dpk}_n \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_i)$.
2. For the h-th query, $\mathcal{Z}$ requests participant P_j to provide a derived public key dpk_h related to mpk_i. $\mathcal{G}_h$ randomly selects a public key mpk_r such that $\mathsf{mpk}_i \neq \mathsf{mpk}_r$ and queries the oracle $\mathcal{O}^{\mathsf{DPKDerive}}(\cdot)$ with the information $(\mathsf{mpk}_i, \mathsf{mpk}_r)$ to obtain the target derived public key dpk_h. Subsequently, $\mathcal{G}_h$ submits dpk_h as the derived public key for mpk_i. In other words, $\mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{PP}, \mathsf{mpk}_i)$ where $b = 0$ or $\mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{PP}, \mathsf{mpk}_r)$ where $b = 1$.
3. For the remaining $l-h$ queries, $\mathcal{Z}$ requests participant P_j to provide a derived public key dpk_n related to mpk_i, where $n \in [l] \backslash [h]$. $\mathcal{G}_h$ instructs P_j to return $\mathsf{dpk}_n \leftarrow \mathsf{DPKDerive}(\mathsf{PP}, \mathsf{mpk}_r)$.
4. Whenever participant P_i is activated with the input $(\mathsf{DPKOwnerCheck}, \mathsf{sid}, \mathsf{dpk})$, $\mathcal{G}_h$ instructs P_i to return the corresponding result f, where $f = 1$ indicates that dpk is linked to the public key of P_i. Otherwise, $\mathcal{G}_h$ queries the oracle $\mathcal{O}^{\mathsf{DPKAdd}}(\cdot, \cdot)$ about

dpk and receives the result value f, instructing P_i to return this value to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.

5. Whenever participant P_i is activated with the input $(\mathsf{Sign}, \mathsf{sid}, M, R, \mathsf{dpk})$, $\mathcal{G}_h$ instructs P_i to output the execution result $(\mathsf{Signature}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk})$ and sends it to $\mathcal{Z}$. Otherwise, $\mathcal{G}_h$ queries the oracle $\mathcal{O}^{\mathsf{Sign}}(\cdot,\cdot,\cdot)$ with dpk and (M, R), receiving the corresponding signature σ. $\mathcal{G}_h$ instructs P_i to return the information $(\mathsf{Sign}, \mathsf{sid}, M, R, \sigma, \mathsf{dpk})$ to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
6. Whenever participant P_j is activated with the input $(\mathsf{Verify}, \mathsf{sid}, M, R, \sigma)$, $\mathcal{G}_h$ instructs P_j to output the execution result $(\mathsf{Verified}, \mathsf{sid}, M, R, \sigma, f)$ to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
7. Whenever participant P_j is activated with the input $(\mathsf{Link}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1)$, $\mathcal{G}_h$ instructs P_j to output the execution result $(\mathsf{Linked}, \mathsf{sid}, M_0, R_0, \sigma_0, M_1, R_1, \sigma_1, f)$ to $\mathcal{Z}$. This is a perfect simulation, and at this step, $\mathcal{Z}$ cannot distinguish between $(\mathsf{S}, \mathcal{F}_{\mathsf{SALRS}})$ and $(\mathcal{A}, \pi_\Sigma)$.
8. When $\mathcal{Z}$ halts, $\mathcal{G}_h$ outputs the output value of $\mathcal{Z}$ and halts.

We analyze the success probability of $\mathcal{G}_h$ using the methodology of hybrid argument. For $j \in \{0, \ldots, l\}$, let $\mathbf{Env}_j$ represent the event: $\mathcal{Z}$ interacts with S in the ideal world, except that the first j derived public keys are derived from the real master public key mpk_i instead of mpk_r. Let H_j be $\Pr\left[\mathcal{Z} \to 1 | \mathbf{Env}_j\right]$.

We easily observe that H_0 is equivalent to the probability of $\mathcal{Z}$ outputting 1 in the ideal world, and H_l is equivalent to the probability of $\mathcal{Z}$ outputting 1 in the real world. Moreover, during the execution of $\mathcal{G}_h$, if $\mathcal{G}_h$ obtains the value dpk_h from its derived public key oracle, where dpk_h is derived from the genuine master public key mpk_i, then the probability of $\mathcal{Z}$ outputting 1 is equivalent to H_h. If dpk_h is derived from the master public key mpk_r, then the probability of $\mathcal{Z}$ outputting 1 is equivalent to H_{h-1}. The detailed process is as follows:

$$\begin{aligned}
H_0 &= \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{SALRS}}, \mathsf{S}, \mathcal{Z}}(\kappa, z) \\
H_l &= \mathsf{REAL}_{\pi_\Sigma, \mathcal{A}, \mathcal{Z}}(\kappa, z) \\
H_{h-1} &= \Pr[\mathcal{G}_h \to 1 | \mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_r)] \\
H_h &= \Pr[\mathcal{G}_h \to 1 | \mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_i)].
\end{aligned} \tag{7}$$

$$\begin{aligned}
\sum_{i=1}^{l} |H_{i-1} - H_i| \geq \left| \sum_{i=1}^{l} (H_{i-1} - H_i) \right| &= |H_0 - H_l| \\
&= \left| \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{SALRS}}, \mathsf{S}, \mathcal{Z}}(\kappa, z) - \mathsf{REAL}_{\pi_\Sigma, \mathcal{A}, \mathcal{Z}}(\kappa, z) \right| > \mathsf{negl}(\kappa).
\end{aligned} \tag{8}$$

Similar to the proof of signer-anonymity, there exists some $h \in \{0, \ldots, l\}$ such that $|H_{h-1} - H_h| > \frac{\mathsf{negl}(\kappa)}{l}$. Here, without loss of generality, we assume $H_{h-1} - H_h > \frac{\mathsf{negl}(\kappa)}{l}$. Thus, the advantage of the adversary $\mathcal{G}_h$ is as follows:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{mpkunl}}_{\Sigma, \mathcal{G}_h}(\kappa) =& \Pr\left[\mathbf{Exp}^{\mathsf{mpkunl},1}_{\Sigma, \mathcal{A}}(\kappa) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{mpkunl},0}_{\Sigma, \mathcal{A}}(\kappa) = 1\right] \\
=& \Pr[\mathcal{G}_h \to 1 | \mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_i)] \\
& - \Pr[\mathcal{G}_h \to 1 | \mathsf{dpk}_h \leftarrow \mathsf{DPKDerive}(\mathsf{mpk}_r)] \\
=& H_h - H_{h-1} > \frac{\mathsf{negl}(\kappa)}{l}
\end{aligned} \tag{9}$$

This implies that $\mathcal{G}$ has a non-negligible advantage with respect to κ, as l is polynomially bounded in κ. Therefore, if the environment $\mathcal{Z}$ can distinguish between the real and ideal worlds, there exists an adversary $\mathcal{B}$ that, under the help of the environment $\mathcal{Z}$, breaks the master-public-key-unlinkability of Σ. □

Consequently, we arrive at the following theorem.

Theorem 1. *Let Σ be an SALRS scheme. The corresponding protocol π_{Σ} securely realizes the ideal functionality $\mathcal{F}_{SALRS}$ if and only if the scheme Σ satisfies signer-linkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability simultaneously.*

Proof. The proof can be deduced from the preceding two lemmas. □

5. Conclusions

In this paper, we revisited and formalized the ideal functionality of the linkable ring signature supporting stealth addresses (SALRS) within the universal composability (UC) model, encapsulating all correctness, soundness, and privacy considerations. Furthermore, our research conclusively demonstrates that the newly introduced UC-security feature for SALRS aligns with the simultaneous fulfillment of essential game-based security properties: signer-unlinkability, signer-non-slanderability, signer-anonymity, and master-public-key-unlinkability. This finding not only safeguards the sustained security of pre-existing SALRS designs within the UC framework but also highlights their seamless integration capabilities with other UC-secure primitives in intricate blockchain systems. Future research may focus on providing security proofs for more cryptographic primitives in the UC model within the context of blockchain, thereby strengthening the overall security of the blockchain structure.

Author Contributions: Conceptualization, X.W. and Z.L.; methodology, X.W., C.Z. and Z.L.; formal analysis, X.W.; writing—original draft preparation, X.W.; writing—review and editing, X.W. and Z.L.; supervision, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (No. 62072305, 62132013).

Data Availability Statement: The data will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PPT	Probabilistic Polynomial Time
SALRS	Linkable Ring Signature Supporting Stealth Addresses
UC	Universal Composability

References

1. Van Saberhagen, N. CryptoNote v 2.0. 2013. Available online: https://www.bytecoin.org/old/whitepaper.pdf (accessed on 20 November 2023).
2. Liu, J.K.; Wei, V.K.; Wong, D.S. Linkable spontaneous anonymous group signature for ad hoc groups. In Proceedings of the Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, NSW, Australia, 13–15 July 2004; Proceedings 9; Springer: Berlin/Heidelberg, Germany, 2004; pp. 325–335.
3. Todd, P. Stealth Addresses. Bitcoin Development Mailing List. 6 January 2014. Available online: https://www.mail-archive.com/bitcoin-development@lists.sourceforge.net/msg03613.html (accessed on 20 November 2023).
4. Liu, Z.; Yang, G.; Wong, D.S.; Nguyen, K.; Wang, H. Key-insulated and privacy-preserving signature scheme with publicly derived public key. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 17–19 June 2019; IEEE: New York, NY, USA, 2019; pp. 215–230.
5. Baum, C.; Lin, H.; Oechsner, S. Towards practical lattice-based one-time linkable ring signatures. In Proceedings of the International Conference on Information and Communications Security, Lille, France, 29–31 October 2018; Springer: Cham, Switzerland, 2018; pp. 303–322.
6. Boyen, X.; Haines, T. Forward-secure linkable ring signatures from bilinear maps. *Cryptography* **2018**, *2*, 35. [CrossRef]

7. Branco, P.; Mateus, P. A code-based linkable ring signature scheme. In Proceedings of the Provable Security: 12th International Conference, ProvSec 2018, Jeju, Republic of Korea, 25–28 October 2018; Proceedings 12; Springer: Cham, Switzerland, 2018; pp. 203–219.
8. Courtois, N.T.; Mercer, R. Stealth address and key management techniques in blockchain systems. In Proceedings of the ICISSP 2017—3rd International Conference on Information Systems Security and Privacy, Porto, Portugal, 19–21 February 2017; pp. 559–566.
9. Noether, S.; Mackenzie, A.; Monero Research Lab. Ring confidential transactions. *Ledger* **2016**, *1*, 1–18. [CrossRef]
10. Fujisaki, E. Sub-linear size traceable ring signatures without random oracles. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2012**, *95*, 151–166. [CrossRef]
11. Liu, J.K.; Au, M.H.; Susilo, W.; Zhou, J. Linkable ring signature with unconditional anonymity. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 157–165. [CrossRef]
12. Tsang, P.P.; Wei, V.K. Short linkable ring signatures for e-voting, e-cash and attestation. In Proceedings of the International Conference on Information Security Practice and Experience, Singapore, 11–14 April 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 48–60.
13. Liu, Z.; Nguyen, K.; Yang, G.; Wang, H.; Wong, D.S. A lattice-based linkable ring signature supporting stealth addresses. In Proceedings of the Computer Security—ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, 23–27 September 2019; Proceedings, Part I 24; Springer: Cham, Switzerland, 2019; pp. 726–746.
14. Alberto Torres, W.A.; Steinfeld, R.; Sakzad, A.; Liu, J.K.; Kuchta, V.; Bhattacharjee, N.; Au, M.H.; Cheng, J. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1. 0). In Proceedings of the Information Security and Privacy: 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, 11–13 July 2018; Proceedings 23; Springer: Cham, Switzerland, 2018; pp. 558–576.
15. Zhang, H.; Zhang, F.; Tian, H.; Au, M.H. Anonymous post-quantum cryptocash. In Proceedings of the International Conference on Financial Cryptography and Data Security, Nieuwpoort, Curaçao, 26 February–2 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 461–479.
16. Lu, X.; Au, M.H.; Zhang, Z. Raptor: A practical lattice-based (linkable) ring signature. In Proceedings of the Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, 5–7 June 2019; Proceedings 17; Springer: Cham, Switzerland, 2019; pp. 110–130.
17. Libert, B.; Ling, S.; Nguyen, K.; Wang, H. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Proceedings of the Advances in Cryptology—EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016; Proceedings, Part II 35; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–31.
18. Esgin, M.F.; Steinfeld, R.; Sakzad, A.; Liu, J.K.; Liu, D. Short lattice-based one-out-of-many proofs and applications to ring signatures. In Proceedings of the Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, 5–7 June 2019; Proceedings 17; Springer: Cham, Switzerland, 2019; pp. 67–88.
19. Canetti, R. Universally composable security: A new paradigm for cryptographic protocols. In Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, Newport Beach, CA, USA, 7 August 2002; IEEE: New York, NY, USA, 2002; pp. 136–145.
20. Backes, M.; Hofheinz, D. How to break and repair a universally composable signature functionality. In Proceedings of the International Conference on Information Security, Palo Alto, CA, USA, 27–29 September 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 61–72.
21. Canetti, R. Universally composable signature, certification, and authentication. In Proceedings of the 17th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, USA, 30 June 2004; IEEE: New York, NY, USA, 2004; pp. 219–233.
22. Abe, M.; Ohkubo, M. A framework for universally composable non-committing blind signatures. *Int. J. Appl. Cryptogr.* **2012**, *2*, 229–249. [CrossRef]
23. Hong, X.; Gao, J.; Pan, J.; Zhang, B. Universally composable secure proxy re-signature scheme with effective calculation. *Clust. Comput.* **2019**, *22*, 10075–10084. [CrossRef]
24. Zhu, C.; Wang, X.; Liu, Z. Universally Composable Key-Insulated and Privacy-Preserving Signature Scheme with Publicly Derived Public Key. In Proceedings of the Inscrypt 2023, HangZhou, China, 11–12 November 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.