Anonymous Post-Quantum Cryptocash*

Huang Zhang[1,2], Fangguo Zhang[1,2,**], Haibo Tian[1,2], and Man Ho Au[3]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[2] Guangdong Key Laboratory of Information Security, Guangzhou 510006, China
[3] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract. In this paper, we extend our work in FC 2018 to construct an anonymous and decentralized cryptocash system which is potentially secure against quantum computers. In order to achieve that, we modify the the old linkable ring signature based on ideal-lattices to reduce the key size and give a complete security proof to it. The size of a signature in our scheme is $O(\frac{n \log \mu}{\log(2n) - \log \tau})$, where μ is the cardinality of the ring, n is the security parameter, and $\tau = \log \mu$. The framework of our cryptocash system follows that of CryptoNote with some modifications. By adopting the short quantum-resistant linkable ring signature scheme, our system is anonymous and efficient. We also introduce how to generate the verifying and signing key pairs of the linkable ring signature temporarily. With these techniques, the privacy of users is protected, even though their transactions are recorded in the public ledger.

1 Introduction

Electronic currencies or cryptocash systems have been proposed for many years. But none of them is prevalent before the Bitcoin system appears. Bitcoin was first described by Satoshi Nakamoto in 2008 [29]. Its success is partially due to its properties of decentralization and anonymity. To prevent "double spending", the system maintains the history of transactions among most nodes in a peer-to-peer network. A consensus mechanism called proof-of-work is used to maintain the history.

Later, researchers found that the public history of Bitcoin causes weaknesses which violate its original designing goals. The latest result stated that Bitcoin only addresses the anonymity and unlinkability issues partially [3]. For example, multiple public keys of the same user can potentially be linked when a user sends change back to his wallet. In this case, two or more of a single user's public keys will appear in the same transaction [33]. Recently, there are more discussions about the weak anonymity of Bitcoin [31, 35]. Although this weakness can be

* The paper is the advanced and full version of the paper accepted by FC 2018

** Corresponding author, isszhfg@mail.sysu.edu.cn

overcome by adopting mixing and distributed methods, the solutions have to include a trusted third party which is a violation to the decentralization property.

There are some creative works to design a strong anonymous cryptocash system. Zerocoin [27] and its advanced version Zerocash [37] allow users to spend their coins using anonymous proof of ownership instead of explicit public-key based digital signatures. This is intuitively the most secure but somewhat inefficient approach. For users who are willing to have better efficiency at the cost of relatively weaker anonymity, they could instead use a ring-signature-based cryptocash system, such as CryptoNote [36] and Monero. CryptoNote presented two properties, namely, "untraceability" and "unlinkability", that must be possessed in a fully anonymous cryptocash model. In CryptoNote, to provide anonymity, there are two ways for all transactions on the network: 1) hiding the sender's address using ring signatures, 2) hiding the receiver's identity using stealth addresses. Both sending and receiving addresses are verifying keys of a ring signature scheme.

The notion of ring signatures, introduced by Rivest *et al.* [34], permits a user to sign a message on behalf of a group. A verifier is convinced that the real signer is a member of the group, but cannot explicitly identify the real signer. Nevertheless, to be employed in a cryptocash system, a ring signature has to be transformed into a linkable ring signature [19], so that double spending could at least be detected. This is what has been done in CryptoNote and Monero.

Another hint to develop cryptocash systems comes from the risks brought by quantum computers. Researchers have shown that a quantum algorithm is able to solve number-theoretic problems, such as the discrete logarithm problem (DLP) efficiently, so that cryptocash systems based on them are not secure under the quantum computing model. One solution is to build schemes on computational problems that remain even hard for quantum computers. Lattice problems have been widely believed as a suitable candidate since Ajtai proposed his seminal work [2]. Some post-quantum signature schemes have been proposed recently [9,12,20]. Relying on these schemes, it is easy to obtain a post-quantum cryptocash system by replacing the ECDSA signature scheme in Bitcoin. However, the resulting cryptocash system is simply like Bitcoin in which the transactions are still linkable. So, it is worthwhile to equip a cryptocash system with lattice-based (linkable) ring signatures.

Considering the anonymity of a cryptocash system, a (linkable) ring signature is obviously more suitable than a standard signature. But there is a cost: the size of the signature and the computational complexity are inherently larger than those of a standard signature. A traditional ring signature scheme usually features a signature size of $O(\mu)$, where μ is the cardinality of the ring. To construct a ring signature of $O(\log \mu)$ or $O(1)$ size was an open problem in this field. Currently, there are two $O(\log \mu)$ ring signatures based on number-theoretic assumptions without trusted setup [7, 14]. On the side of lattices, Libert *et al.* have proposed an $O(\log \mu)$ lattice-based ring signature [17], and Yang *et al.* based on [17] designed a linkable ring signature [41].

However, as pointed out in [10], the current shortest logarithmic size ring signature from lattices by Libert *et al.* [17] is not optimal, because in a practical point of view, the size of a signature is not small enough when the number of members is not very large (*e.g.*, 59166KB signature size; 2^{10} ring members; 100 security bits). It is still worthwhile to construct lattice-based ring signatures using the ideas in [7, 14], since those two signatures feature brilliant efficiency. Inspired by this, Esgin *et al.* proposed a lattice-based ring signature scheme [10] by combining the techniques in [7,14]. They also employed some new techniques to handle the problems only caused by lattices. Even though the theoretical size of the signature is larger than $O(\log \mu)$, the concrete efficiency analysis showed that the signature size of [10] is better than that of [17] (*e.g.*, 1409KB signature size; 2^{10} ring members; 100 security bits).

In this paper, we aim at designing an anonymous post-quantum cryptocash (APQC) system. In order to achieve this goal, we propose a linkable ring signature based on ideal-lattices. The size of a signature is $O(\frac{n \log \mu}{\log(2n) - \log \tau})$, where μ is the cardinality of the ring, n is the security parameter, and $\tau = \log \mu$. The framework of our cryptocash system follows that of CryptoNote [36], and the ideal-lattice-based signature scheme is inspired by the work of [14] with some modifications. We notice that, by using the ideas in [7] to commit a bunch of bits simultaneously, we will obtain a linkable ring signature scheme of better signature size comparing to the scheme in this paper. But, as the current work is an extension of our conference paper (FC2018), we mainly focus on giving a completely security proof and reducing key size in the old settings with the techniques suggested in [10]. For the details of constructing a lattice-based ring signature by combining the ideas of [7,14], we refer the readers to [10]. A major distinction between [10] and the current work is that the former employed the Module-SIS problem, and the latter adopted ring-SIS problem. Accordingly, the hardness assumptions and the discussion on statistical distances are different.

The paper is organized as follows: in Sect. 2, we introduce notations and concepts applied in our work. The model of the ring signature based cryptocash is described in Sect. 3. We present a lattice-based one-out-of-many protocol for commitments in Sect. 4. Section 5 involves the concrete construction of the ideal-lattice-based linkable ring signature. We design the standard transaction of our cryptocash system in Sect. 6. Section 7 is a brief conclusion for this paper.

2 Preliminaries

2.1 Notations

We use $\mathbb{Z}$, $\mathbb{Z}^+$, $\mathbb{N}$, $\mathbb{R}$ to denote the set of all integers, the set of all positive integers, the set of all natural numbers, and the set of all reals, respectively. If $a, b \in \mathbb{Z}$ and $a < b$, then $[a, b]$ is the set $\{x \in \mathbb{Z} : a \leq x \leq b\}$ and $[a, b)$ is the set $\{x \in \mathbb{Z} : a \leq x < b\}$. For an integer i, i_j symbolizes the j-th bit of i. $\delta_{i\ell}$ is Kronecker's delta, *i.e.*, $\delta_{\ell\ell} = 1$ and $\delta_{i\ell} = 0$ for $i \neq \ell$. An ordered list $(x_1, \ldots, x_n)$ is denoted by $(x_i)_{i=1}^n$. We use $|S|$ to indicate the cardinality of a set finite S, and

$a \leftarrow S$ means a is chosen from S uniformly at random. But if D is a distribution, $a \leftarrow D$ implies that a is sampled according to D. For two strings (or vectors) $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_m)$, $(a\|b) = (a_1, \ldots, a_n, b_1, \ldots, b_m)$ is their concatenation.

For $n, q \in \mathbb{Z}^+$, let $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ in the remaining of the paper. An element in R_q is a polynomial of the form $a = a_0X^0 + \cdots + a_{n-1}X^{n-1}$, so that it can also be denoted by its coefficients, written as $\mathrm{vec}(a) = (a_0, \ldots, a_{n-1}) \in \mathbb{Z}_q^n$. Sometimes we will abuse these notations for simplicity. Let $Rot(a)$ be the $n \times n$ matrix consisting of the vectors $(\mathrm{vec}(aX^0), \ldots, \mathrm{vec}(aX^{n-1}))$, where $\mathrm{vec}(aX^i)$ is parsed as a column vector. For a vector $a \in \mathbb{Z}^n$, $\|a\|_p$ represents its ℓ_p norm, and p is omitted if $p = 2$. The norm of a vector $\mathbf{a} = (a_1, \ldots, a_m) \in R_q^m$ is measured by regarding it as a vector $(\mathrm{vec}(a_1)\| \ldots \|\mathrm{vec}(a_m)) \in \mathbb{Z}_q^{nm}$. If $a \in R_q$ and $\mathbf{x} \in R_q^m$, then $a \cdot \mathbf{x}$ (or sometimes $a\mathbf{x}$) denotes the scalar multiplication. For two vectors $\mathbf{a}, \mathbf{b} \in R_q^m$, $\langle \mathbf{a}, \mathbf{b}\rangle$ is their inner product.

If a function $f : \mathbb{N} \to \mathbb{N}$ vanishes faster than the reciprocal of any positive polynomial function $\mathrm{poly} : \mathbb{N} \to \mathbb{N}$, we say f is negligible, written $f(n) = \mathrm{negl}(n)$. On the other side, we call $1 - \mathrm{negl}(n)$ the overwhelming probability.

2.2 Lattices and Theories

A lattice $\Lambda = \mathcal{L}(\mathbf{B})$ with dimension m and rank n is a subgroup of the linear space $\mathbb{R}^m$. Every element in Λ can be represented as an integral combination of its basis $\mathbf{B} \in \mathbb{R}^{m \times n}$. A lattice that corresponding to a parity check matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is defined by $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$. In our work, we will focus on a specific class of lattices, called ideal lattices, which can be described as ideals of certain polynomial rings.

Definition 1 (Definition 2, [21]). *An ideal lattice is an integer lattice $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^n$ such that $\mathcal{L}(\mathbf{B}) = \{g \bmod f : g \in \mathcal{I}\}$ for some monic polynomial f of degree n and ideal $\mathcal{I} \in \mathbb{Z}[X]/\langle f\rangle$.*

To extend the hash function family in previous works [2, 8, 25], Micciancio defined the generalized knapsack function family [23, 24].

Definition 2 (Definition 4.1, [24]). *For any ring R, subset $D \subset R$ and integer $m \geq 1$, the generalized knapsack function family $\mathcal{K}(R, D, m) = \{f_{\mathbf{a}} : D^m \to R\}_{\mathbf{a} \in R^m}$ is defined by*

$$f_{\mathbf{a}}(\mathbf{x}) = \sum_{i=1}^{m} x_i \cdot a_i \ ,$$

for all $\mathbf{a} \in R^m$ and $\mathbf{x} \in D^m$, where $\sum_i x_i \cdot a_i$ is computed using the ring addition and multiplication operations.

If $R = R_q$, the computation of $f_{\mathbf{a}}(\mathbf{x})$ can be rewritten as a matrix-vector product $(\mathbf{A}^{(1)}\| \ldots \|\mathbf{A}^{(m)}) \cdot \bar{\mathbf{x}}^T$ in $\mathbb{Z}_q$, where $\mathbf{A}^{(i)} = Rot(a_i)$, $\bar{\mathbf{x}} = (\mathrm{vec}(x_1)\| \ldots \|\mathrm{vec}(x_m))$. See Sect 4.2, [26] for details. This form shows the relation between a general lattice and an ideal lattice.

Besides one-wayness, Micciancio showed that for a special case of the above function family, the distribution of $f_{\mathbf{a}}(\mathbf{x})$ is uniform and independent from $\mathbf{a}$.

Theorem 1 (Theorem 4.2, [24]). *For any finite field $\mathbb{F}$, subset $S \subset \mathbb{F}$, and integers n, m, the hash function family $\mathcal{K}(\mathbb{F}^n, S^n, m)$ is ϵ-regular for*

$$\epsilon = \frac{1}{2}\sqrt{(1 + |\mathbb{F}|/|S|^m)^n - 1} \ .$$

In particular, for any $q = n^{O(1)}$, $|S| = n^{\Omega(1)}$ and $m = \omega(1)$, the function ensemble $\mathcal{K}(\mathbb{F}_q^n, S^n, m)$ is almost regular (i.e., $\epsilon(n) = \mathrm{negl}(n)$).

Here, "ϵ-regular" means that the statistical distance between uniform distribution $U((\mathbb{F}^n)^m, \mathbb{F}^n)$ and $\{(\mathbf{a}, f_{\mathbf{a}}(\mathbf{x})) : \mathbf{a} \leftarrow U((\mathbb{F}^n)^m), \mathbf{x} \leftarrow U((S^n)^m)\}$ is at most ϵ. $\mathbb{F}^n$ is a ring of n-dimensional vectors with the usual vector addition operation and convolution product, so that if $\mathbb{F}_q = \mathbb{Z}_q$, we have $\mathbb{F}_q^n$ is isomorphic to R_q.

Sometimes, one-wayness is not sufficient enough for the design of a cryptographic protocol. Lyubashevsky and Micciancio proved that finding a collision in some instance of the generalized knapsack function family is as hard as solving the worst-case problem in a certain lattice [21].

Definition 3 (Collision Problem). *For any generalized knapsack function family $\mathcal{K}(R, D, m)$, define the collision problem $\mathrm{Col}_{\mathcal{K}}(h_{\mathbf{a}})$ as follows: given a function $h_{\mathbf{a}} \in \mathcal{K}$, find $\mathbf{b}, \mathbf{c} \in D^m$ such that $\mathbf{b} \neq \mathbf{c}$ and $h_{\mathbf{a}}(\mathbf{b}) = h_{\mathbf{a}}(\mathbf{c})$.*

If there is no polynomial time algorithm that can solve $\mathrm{Col}_{\mathcal{K}}$ with non-negligible probability when given a function $h_{\mathbf{a}}$ which is uniformly chosen from $\mathcal{K}$ at random, then $\mathcal{K}$ is a collision resistant family of hash functions.

The expansion factor is a parameter proposed to quantify the quality of modulus f in the ideal lattice [21]. The expansion factor of f is defined as

$$\mathrm{EF}(f, k) = \max_{g \in \mathbb{Z}[x], \deg(g) \leq k(\deg(f)-1)} \|g\|_f / \|g\|_\infty$$

where $\|g\|_f$ is short for $\|g \bmod f\|_\infty$. Moreover, $\mathrm{EF}(X^n + 1, k) \leq k$.

The generalized knapsack function family $\mathcal{K}(R, D, m)$ considered in [21] is instantiated as follows. Let $R = \mathbb{Z}_q[X]/\langle f \rangle$ be a ring for some integer q, where $f \in \mathbb{Z}[X]$ is a monic, irreducible polynomial of degree n with expansion factor $\mathrm{EF}(f, 3) \leq \varepsilon$. Let $D = \{g \in R : \|g\| \leq \beta\}$ for some positive integer β. With such a setting, the following theorem shows the hardness of the collision problem.

Theorem 2 (Theorem 2, [21]). *Let $\mathcal{K}(R, D, m)$ be a generalized knapsack function family as above with $m \geq \frac{\log q}{\log 2\beta}$ and $q > 2\varepsilon\beta mn^{1.5} \log n$. Then, for $\gamma = 8\varepsilon^2\beta mn \log^2 n$, there is a polynomial time reduction from f-$\mathrm{SPP}_\gamma(\mathcal{I})$ for any ideal $\mathcal{I} \in R$ to $\mathrm{Col}_{\mathcal{K}}(h)$ where h is chosen uniformly at random from $\mathcal{K}$.*

If we denote by $\mathcal{I}(f)$ the set of lattices that are isomorphic (as additive groups) to ideals of $\mathbb{Z}[X]/(f)$ where f is monic, then there is a straightforward reduction from $\mathcal{I}(f)$-SVP_γ to f-SPP_γ, and the vise versa. It is conjectured that approximating $\mathcal{I}(f)$-SVP_γ to within a polynomial factor is a hard problem, although it is not NP-hard [1, 13].

In the security proof of the current paper, we wish to ensure that a ring element with small norm is invertible. The lemma below was employed in [10] to obtain such a guarantee.

Lemma 1 (Corollary 1.2, [22]). *Let $q \equiv 5 \pmod 8$ be a prime, n be a positive power of 2. Then any non-zero polynomial $z \in R_q$ with $2\|z\|_\infty^2 < q$ or $\|z\|^2 < q$ is invertible in R_q.*

When making algebraic operations in R_q, we have the following results on norms.

Lemma 2 (Lemma 4, [10]). *For $a, b \in R_q$, we have the following relations*

$$\|a\|_\infty \leq \|a\| \leq \sqrt{n} \cdot \|a\|_\infty, \quad \|a \cdot b\| \leq \sqrt{n} \cdot \|a\| \cdot \|b\|, \quad \text{and } \|a \cdot b\|_\infty \leq \|a\| \cdot \|b\| \ .$$

As suggested by [10], we make use of a special challenge space in our Sigma-protocol. Moreover, we require that the Vandermonde matrix consisting of distinct challenges is invertible in R_q.

Lemma 3 (Lemma 6 and Lemma 7, [10]). *Let $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$ where $n > 1$ is a power of 2, and $0 < i, j < 2n - 1$. Then, all the coefficients of $2(X^i - X^j)^{-1} \in R$ are in $\{-1, 0, 1\}$. This implies that $\|2(X^i - X^j)^{-1}\| \leq \sqrt{d}$. Furthermore, for τ distinct elements $(x_0, \ldots, x_\tau)$, where $x_i = X^{w_i} \in R$, and $w_i \in [0, 2n-1]$, the Vandermonde matrix corresponding to the τ distinct elements is invertible in $R_q = R/qR$ for odd q, and if*

$$(\alpha_0, \ldots, \alpha_\tau) = (0, 0, \ldots, 0, 1) \begin{pmatrix} 1 & x_0^1 & x_0^2 & \ldots & x_0^\tau \\ 1 & x_1^1 & x_1^2 & \ldots & x_1^\tau \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_\tau^1 & x_\tau^2 & \ldots & x_\tau^\tau \end{pmatrix}^{-1} \ ,$$

then for $i \in [0, \tau]$, $\|2^\tau \alpha_i\| \leq n^{\tau - 0.5}$.

The discrete Gaussian distribution is widely used in lattice-based cryptography. The following definition comes from [25].

Definition 4. *For any $\sigma > 0$, define the Gaussian function on $\mathbb{R}^m$ centered at $\mathbf{v} \in \mathbb{R}^m$ with parameter σ as*

$$\forall \mathbf{x} \in \mathbb{R}^m, \quad \rho_{\mathbf{v},\sigma}(\mathbf{x}) = e^{-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2} \ .$$

The subscript $\mathbf{v}$ is taken to be $\mathbf{0}$ when omitted. Relying on this, the discrete Gaussian distribution over an m-dimensional lattice Λ centered at $\mathbf{v} \in \mathbb{R}^m$ with Gaussian parameter σ is defined as $\forall \mathbf{x} \in \Lambda, D_{\Lambda,\mathbf{v},\sigma}(\mathbf{x}) = \rho_{\mathbf{v},\sigma}(\mathbf{x}) / \rho_\sigma(\Lambda)$.

Notice that $\mathbb{Z}^m$ is also a lattice. In the rest of paper, the discrete Gaussian distribution over $\mathbb{Z}^m$ with center $\mathbf{v} \in \mathbb{Z}^m$ and parameter $\sigma > 0$ will be denoted by $D^m_{\mathbf{v},\sigma}$.

The smoothing parameter is a lattice parameter related to Gaussian measures on lattice defined in [25].

Definition 5 (Smoothing parameter). *For an n-dimensional lattice Λ, and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest s such that $\rho_{1/\sigma}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$, where Λ^* is the dual lattice of Λ.*

The smoothing parameters are connected with various properties of the discrete Gaussian distribution over lattices. The following two are useful when showing that $\mathcal{CMT}$ in Sect. 4.1 is statistically hiding.

Lemma 4 (Lemma 5.2, full version of [12]). *Assume the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generate $\mathbb{Z}_q^n$, and let $\epsilon \in (0, \frac{1}{2})$ and $\sigma \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$. Then for $\mathbf{e} \leftarrow D_\sigma^m$, the distribution of the syndrome $\mathbf{u} = \mathbf{A}\mathbf{e} \bmod q$ is within statistical distance 2ϵ of uniform over $\mathbb{Z}_q^n$.*

Note that if $\mathbb{F}^n = \mathbb{Z}_q^n = R_q$ and ϵ is a negligible function in Theorem 1, then it implies that a uniformly chosen $\mathbf{a} \in R_q^m$ generates $R_q = \mathbb{Z}_q^n$ with overwhelming probability.

Lemma 5 (Lemma 5.3, full version of [12]). *Let n and q be positive integers with q prime, and let $m \geq 2n \log q$. Then for all but a q^{-n} fraction of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the columns of $\mathbf{A}$ generates $\mathbb{Z}_q^n$. For such $\mathbf{A}$, we have $\lambda_1^\infty \geq \frac{q}{4}$. In particular, for such $\mathbf{A}$ and for any $\omega(\sqrt{\log m})$ function, there is a negligible function $\epsilon(m)$ such that $\eta_\epsilon(\Lambda^\perp(\mathbf{A})) \leq \omega(\sqrt{\log m})$.*

With above two lemmas, we know that if the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generates $\mathbb{Z}_q^n$, and $\sigma \geq \omega(\sqrt{\log m})$, then the distribution of $\mathbf{A}\mathbf{x} \bmod q$ is within negligible distance to uniform, where $\mathbf{x} \leftarrow D_\sigma^m$.

If a random variable in a lattice is sampled according to the discrete Gaussian distribution, the upper-bound of its l_2 norm is predicable with overwhelming probability.

Lemma 6 (Lemma 3.1, full version of [12]). *For any n-dimensional lattice Λ and real $\epsilon > 0$, we have*

$$\eta_\epsilon(\Lambda) \leq \widetilde{bl}(\Lambda) \cdot \sqrt{\log(2n(1 + 1/\epsilon)/\pi)} \ .$$

Then for any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon(n)$ for which $\eta_\epsilon(\Lambda) \leq \widetilde{bl}(\Lambda) \cdot \omega(\sqrt{\log n})$, where $\widetilde{bl}(\Lambda) = \min_{\mathbf{B}} \|\widetilde{\mathbf{B}}\|$ is the Gram-Schmidt minimum norm of the basis of Λ.

Notice that if $\Lambda = \mathbb{Z}_q^n$, then $\eta_\epsilon(\Lambda) \leq \omega(\sqrt{\log n})$, since the identity matrix is a basis for Λ.

Lemma 7 (Lemma 4.4, [25]). *For any n-dimensional lattice Λ, vector $\mathbf{c} \in \mathbb{R}^n$, and reals $0 < \epsilon < 1$, $\sigma \geq \eta_\epsilon(\Lambda)$, we have*

$$\Pr_{\mathbf{x} \leftarrow D_{\Lambda, \mathbf{c}, \sigma}} [\|\mathbf{x} - \mathbf{c}\| \geq s\sqrt{n}] \leq \frac{1 + \epsilon}{1 - \epsilon} \cdot 2^{-n} \ .$$

Combining the above two lemmas, we know that for lattice $\mathbb{Z}^n$, if $\sigma \geq \omega(\sqrt{\log n})$, then $\Pr_{\mathbf{x} \leftarrow D_\sigma^n}[\|\mathbf{x}\| \geq s\sqrt{n}] \leq \mathrm{negl}(n)$.

As the discrete Gaussian distribution over $\mathbb{Z}^m$ leaks information about the center $\mathbf{v}$, the notion of rejection samplings was built to handle this problem.

Lemma 8 (Theorem 3.4, [20]). *Let V be a subset of $\mathbb{Z}^m$ in which all elements have l_2 norms less than T, σ be some element in $\mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists a constant $M = O(1)$ such that the distribution of the following algorithm $\mathcal{A}$:*

1. $\mathbf{v} \leftarrow h$
2. $\mathbf{z} \leftarrow D^m_{\mathbf{v},\sigma}$
3. *output* $(\mathbf{z}, \mathbf{v})$ *with probability* $\min\left\{\frac{D^m_\sigma(\mathbf{z})}{MD^m_{\mathbf{v},\sigma}(\mathbf{z})}, 1\right\}$

is within statistical distance $\frac{2^{-\omega(\log m)}}{M}$ of the distribution of the following algorithm $\mathcal{F}$:

1. $\mathbf{v} \leftarrow h$
2. $\mathbf{z} \leftarrow D^m_\sigma$
3. *output* $(\mathbf{z}, \mathbf{v})$ *with probability* $1/M$

Moreover, the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$.

More concretely, if $\sigma = \alpha T$ for any positive α, then $M = e^{12/\alpha + 1/(2\alpha^2)}$, the output of algorithm $\mathcal{A}$ is within statistical distance $\frac{2^{-100}}{M}$ of the output of $\mathcal{F}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-100}}{M}$.

The splitting lemma translates the fact that when a subset A is "large" in a product space $X \times Y$, it has many "large" sections.

Lemma 9 (Lemma 7, [32]). *Let $A \subset X \times Y$ such that $\Pr_{(x,y) \leftarrow X \times Y}[(x, y) \in A] \geq \varepsilon$. For any $\alpha < \varepsilon$, define*

$$B = \left\{ x \in X : \Pr_{y \leftarrow Y}[(x, y) \in A] \geq \varepsilon - \alpha \right\} ,$$

then the following statements hold:

1. $\Pr_{x \leftarrow X}[x \in B] \geq \alpha$.
2. $\forall x \in B$, $\Pr_{y \leftarrow Y}[(x, y) \in A] \geq \varepsilon - \alpha$.

Using the notations as in the above lemma, but letting $Y = \mathcal{C}^r \times Z$, the authors of [10] proved the lemma below.

Lemma 10 (Claim 1, [10]). *Let $A \subseteq X \times \mathcal{C}^r \times Z$ and*

$$B = \{a \in X : \Pr_{(\mathbf{x},b) \leftarrow \mathcal{C}^r \times Z}[(a, \mathbf{x}, b) \in A] \geq \varepsilon'\} ,$$

where $\mathbf{x}$ *is an* r*-dimensional vector and is denoted by* $\mathbf{x} = (x_1, \ldots, x_r)$*. Conditioned on* $a \in B$*, define the conditional probability regarding to a* $c \in \mathcal{C}$ *by*

$$p_i(c) = \Pr_{(\mathbf{x},b) \leftarrow \mathcal{C}^r \times Z}[(a, \mathbf{x}, b) \in A \wedge x_i = c \mid a \in B] \ .$$

If $\varepsilon' > (\tau/|\mathcal{C}|)^r$*,* $|\mathcal{C}| \geq \tau$*, then there exists an* $i^* \in [1, r]$ *and* $G \subseteq \mathcal{C}$ *with* $|G| = \tau + 1$ *such that*

$$\forall c \in G, \ \ p_{i^*}(c) \geq \frac{\varepsilon' - (\tau/|\mathcal{C}|)^r}{(|\mathcal{C}| - \tau) \cdot r} \stackrel{\text{def}}{=} p \ .$$

2.3 Sigma-Protocol

Let $\mathcal{R}$ be an efficiently decidable ternary relation. If $(crs, u, w) \in \mathcal{R}$, we call u the statement and w the witness, where crs is a common reference string. Let $\mathcal{L}$ be the CRS-dependent language consisting of statements in $\mathcal{R}$. The following definition of Σ-protocols is an extension of the one in [6]. We mainly change it to handle a ternary relation.

Definition 6. *Let* $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *be a two-party protocol, where* V *is PPT,* $\mathcal{G}$ *is a common reference string generation algorithm. Let* $\mathcal{R}, \mathcal{R}'$ *be two efficiently decidable ternary relation such that* $\mathcal{R} \subseteq \mathcal{R}'$*, and* $\mathcal{L}, \mathcal{L}'$ *be their corresponding CRS-dependant languages Then* $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *is called a* Σ'_m*-protocol for* $\mathcal{R}, \mathcal{R}'$ *with challenge set* $\mathcal{C}$*, public input* crs, u *and private input* w*, completeness error* α*, soundness error* $\frac{m-1}{|\mathcal{C}|}$*, if and only if it satisfies the following conditions:*

- ***3-move form****: The protocol is of the following form:*
  - $crs \leftarrow \mathcal{G}(1^\lambda)$*:* $\mathcal{G}$ *produces the common reference string* crs*, on input a security parameter* λ*.*
  - $a \leftarrow \mathcal{P}(crs, u, w)$*:* $\mathcal{P}$ *takes as input* $(crs, u, w) \in \mathcal{R}$ *and generates an initial message* a*.*
  - $x \leftarrow \mathcal{V}(crs, u, a)$*:* $\mathcal{V}$ *sends to* $\mathcal{P}$ *a challenge* $x \in \mathcal{C}$ *chosen uniformly at random.*
  - $z \leftarrow \mathcal{P}(crs, u, w, x)$*: On input* x*,* $\mathcal{P}$ *gives a response* z *to* $\mathcal{V}$ *in return.*
  - 0 *or* $1 \leftarrow \mathcal{V}(crs, u, a, x, z)$*: Given* (crs, u, a, x, z)*,* $\mathcal{V}$ *returns* 1 *if accepting the proof and 0 if rejecting the proof.*
- ***Completeness****: Whenever* $(crs, u, w) \in \mathcal{R}$*, the* $\mathcal{V}$ *accepts with probability at least* $1 - \alpha$*.*
- **m-*Special soundness****: There exists a PPT algorithm* $\mathcal{X}$ *(the knowledge extractor) which takes* m *accepting transcripts* $(a, x_1, z_1), \ldots, (a, x_m, z_m)$ *satisfying* $x_i \neq x_j$ *for* $i \neq j$ *as input, and outputs* w' *such that* $(crs, u, w') \in \mathcal{R}'$*.*
- ***Special honest-verifier zero-knowledge (SHVZK)****: There exists a PPT algorithm* $\mathcal{S}$ *(the simulator) taking* $u \in \mathcal{L}$ *and* $x \in \mathcal{C}$ *as inputs, that outputs transcript* (a, x, z) *whose distribution is (computationally) indistinguishable from accepting protocol transcripts generated by real protocol runs.*

2.4 Linkable Ring Signature

A linkable ring signature consists of five efficient algorithms (**Setup**, **KGen**, **Sign**, **Vfy**, **Link**).

- $pp \leftarrow \mathbf{Setup}(1^\lambda)$: On input a security parameter λ, the algorithm generates and publishes the system parameters pp. pp will be the default input of the other algorithms. We denote by $\mathcal{EID}$, $\mathcal{M}$ the domains of event-id and messages, respectively.
- $(vk, sk) \leftarrow \mathbf{KGen}(pp)$: This algorithm generates a verifying and signing key pair (vk, sk).
- $\sigma \leftarrow \mathbf{Sign}(sk, \text{msg}, event, L)$: Output a signature σ on the message msg with respect to the ring L and the event-id $event \in \mathcal{EID}$. It is required that the verifying key vk corresponding to sk is in L.
- $\{0,1\} \leftarrow \mathbf{Vfy}(\text{msg}, event, L, \sigma)$: Verify a purported signature σ on a message msg with respect to the ring L and the event description $event$. It outputs 1 if accepting and 0 if rejecting the signature.
- $\{0,1\} \leftarrow \mathbf{Link}(event, \text{msg}_1, L_1, \sigma_1, \text{msg}_2, L_2, \sigma_2)$: On input two accepting signatures σ_1, σ_2 on the same event description $event$, output 1 if the signatures are linked, and output 0 otherwise.

Definition 7. *A linkable ring signature scheme is of statistically correctness, if any* $pp \leftarrow \mathbf{Setup}(1^\lambda)$, $(vk, sk) \leftarrow \mathbf{KGen}(pp)$, *any* L *such that* $vk \in L$, $event \leftarrow \mathcal{EID}$, $\text{msg} \leftarrow \mathcal{M}$, *and* $\sigma \leftarrow \mathbf{Sign}(sk, \text{msg}, event, L)$.

$$\Pr[\mathbf{Vfy}(\text{msg}, event, \sigma, L) = 1] = 1 - \text{negl}(\lambda) \ .$$

Security of linkable ring signature schemes involves unforgeability, anonymity, linkability, and nonsalnderbility. We employs the definitions proposed in [18] except for the anonymity. As a result, we only explicitly give the definition of statistically weak anonymity in the current paper, but refer the readers to [18] for the details of the other definitions.

There are four oracles in defining these concepts in a random oracle model.

- $vk_i \leftarrow \mathcal{JO}(\perp)$: On request, the *Joining Oracle* runs $(vk_i, sk_i) \leftarrow \mathbf{KGen}(pp)$ with fresh random coins, and returns vk_i.
- $sk_i \leftarrow \mathcal{CO}(vk_i)$: The *Corruption Oracle*, on input a verifying key vk_i which is generated by $\mathcal{JO}$, returns the corresponding signing key sk_i.
- $\sigma \leftarrow \mathcal{SO}(vk_i, \text{msg}, event, L)$: The *Signing Oracle*, on input an $event \in \mathcal{EID}$, a ring L, the verifying key of the signer vk_i such that vk_i is generated by $\mathcal{JO}$, and a message $\text{msg} \in \mathcal{M}$, returns a valid signature σ
- $\mathbf{y} \leftarrow \mathcal{RO}(\mathbf{x})$: On input an element from the domain of the random oracle, returns the corresponding element in the range.

Weak anonymity requires that any party cannot know the actual signer of a ring signature, if all of the parties of the ring do not reveal their identity. Such a property is defined in the following game between the Simulator $\mathcal{S}$ and the PPT adversary $\mathcal{A}$

– $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters pp.
– $\mathcal{A}$ may query $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$, and $\mathcal{RO}$ with arbitrary strategies.
– $\mathcal{A}$ gives $\mathcal{S}$ an event-id $event \in \mathcal{EID}$, a message msg $\in \mathcal{M}$, and a group L such that all of the verifying keys in L are query outputs of $\mathcal{JO}$, and were not submitted to $\mathcal{CO}$.
– $\mathcal{S}$ randomly picks $i \in [1, |L|]$ and computes $\sigma_i = \mathbf{Sign}(sk_i, \text{msg}, event, L)$, where sk_i is the corresponding signing key of vk_i. σ_i is given to $\mathcal{A}$.
– $\mathcal{A}$ outputs a guess $i' \in [1, |L|]$.

We denote $\mathcal{A}$'s advantage in winning this game by

$$\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = \left| \Pr[i' = i] - \frac{1}{|L|} \right| .$$

Definition 8 (Anonymity). *A linkable ring signature scheme is of statistically weak anonymity if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = \text{negl}(\lambda)$.*

3 Anonymous Cryptographic Currency Model Based on (Linkable) Ring Signatures

Cryptocash system based on linkable ring signatures emerged after researchers found that Bitcoin was not fully anonymous and untraceable. CryptoNote and Monero are two typical instances. We describe here the properties of an anonymous cryptocash system and state the techniques [36] to construct such a system.

In a cryptocash system, there are three parties: a sender, who owns a coin and decides to spend it, a receiver, who is the destination that a coin is delivered to, and a public ledger where all transactions are recorded. An anonymous cryptocash system should satisfy the following properties:

– **Untraceability**: If Tx is a transaction from sender A to receiver B, and Tx has been recorded in the public ledger, no one else can determine the sender with probability significantly larger than $1/\mu$ by accessing the transcript of Tx, where μ is the number of possible senders in a related input of the Tx. Moreover, even receiver B cannot prove that A is the true sender of Tx.
– **Unlinkability**: If Tx_1 is a transaction from sender A to receiver C, Tx_2 is another transaction from sender B to receiver C, and Tx_1, Tx_2 have been recorded in the public ledger, then for any subsequent transactions in the public ledger, no one else can use them to link the outputs of Tx_1 and Tx_2 to a single user, even for senders A and B.
– **Detecting Double Spending**: If Tx_1 is a transaction which describes that coin c has been sent from sender A to receiver B, and Tx_1 has been recorded in the public ledger, every user of the system could detect another transaction Tx_2 that describes the same coin c. Furthermore, Tx_2 will never be accepted and recorded in the public ledger.

To design a cryptocash protocol which provides all the above properties, the CryptoNote and Monero suggested to adopt the modification of the traceable ring signature [11], which generates a one-time signature on behalf of a temporal group. Since it is a one-time signature with an explicit identification tag about the signing key, it could prevent a coin being double-spent. Besides, since it is a ring signature where the identity of the real signer is hidden within a set of possible signers, it guarantees untraceability. In addition, ring signature supports unlinkability since the inputs in a transaction may be brought from outputs of transactions belonging to other users.

To employ a linkable ring signature in a cryptocash system, the receiver should produce a one-time key pair for each transaction. A sender could obtain the public key of the receiver for the transaction and build a transaction with an output script containing that key's information. The drawback of this trivial method is that a receiver has to maintain a lot of one-time keys. Furthermore, a sender has to contact each receiver for their fresh one-time public key when the sender builds a transaction. Alternatively, CryptoNote suggests another method which enables a receiver to store only a single key pair. A sender could produce a random value to generate a one-time public key for the receiver based on this single public key. The one-time public key is referred to as the destination address. This is a convenient design at the cost of a slightly weakened unlinkability. Specifically, if a user has a single key, a sender could always identify a receiver from the sender's transaction by its random value of the transaction. If two senders collude, and they have sent coins to the same receiver, they could identify the same receiver while the trivial method avoids this. And if a later transaction includes the two senders' outputs at the same time, with a higher probability, the later transaction is made by the receiver. Note that a receiver could still produce another key pair at will as in the Bitcoin system to avoid the small problem.

Finally, let us observe a standard transaction in a linkable ring signature based cryptocash system. In such a system, the value of a coin is bound with a destination address. Suppose A and B are two users in the system. B has a single key pair (pk_B, sk_B). A has the private key sk_1 of a destination address vk_1, which represents a coin, say c, which has been sent to A previously. If A decides to send c to B, A generates a destination address vk_2 and an auxiliary input aux for B; A then chooses a number of transactions from the public ledger such that the delivered value of coin is equivalent to c; he/she extracts the destination addresses of those transactions and assembles them with vk_1 to form a ring L; A runs a linkable ring signature algorithm to sign transaction Tx, which involves information about (c, aux, vk_2, L), with signing key sk_1 and broadcasts the transaction; If the signature generated by sk_1 is not linkable to any signature on the ledger, the public ledger will accept this transaction and record it; B uses its private key sk_B to check every passing transaction to determine if transaction Tx is for B and recovers the signing key sk_2 corresponding to vk_2. With sk_2, user B can spend c by signing another transaction. However, even A does not

know when and where B spends it due to the functionality of the linkable ring signature.

4 Sigma-Protocols for Commitments

In [14], Groth and Kohlweiss proposed an efficient Sigma-protocol which is responsible for proving that one of μ homomorphic commitments is opened to 0, and the corresponding ring signature is obtained by applying the Fiat-Shamir heuristic to the Sigma-protocol. Our scheme follows their strategy so that in this section, we aim at constructing such a Sigma-protocol.

Indeed, a lattice-based scheme and a DLP-based one share some similarities, but as pointed out in [4, 10, 17], to implement a DLP-based idea with lattice settings is not trivially direct. Variety of specific techniques should be involved to handle the problems caused just by lattices, especially in security proofs. To construct the protocols and to prove their security, we adopt the techniques suggested in [10].

4.1 Commitment Scheme

A non-interactive commitment scheme allows a user to construct a commitment to a value. The user may later open the commitment so any party can check if the opened value is the one that was committed at the beginning.

The non-interactive commitment scheme adopted in current paper consists of a pair of efficient algorithms $\mathcal{CMT} = (\mathbf{Gen}, \mathbf{Com})$.

Table 1. Parameter settings for $\mathcal{CMT}$

Symbol	Setting	Explanation
n	$n = 2^k$, $k \in \mathbb{Z}^+$	$X^n + 1$ is irreducible over integers
m_1	$m_1 \in \mathbb{Z}^+$	length of a message
m_2	$m_2 = \omega(1)$	requirement in Theorem 1
$\mathcal{B}$	$2\mathcal{B} + 1 = n^{\Omega(1)}$	requirement in Theorem 1
σ	$\sigma \geq \omega(\sqrt{\log(m_2 n)})$	requirement in Lemma 4 and 7
β	$\beta \geq \sigma\sqrt{m_2 n}$	valid norm in Theorem 2
	$\beta \geq \mathcal{B}$	valid norm in Theorem 2
q	$q = n^{O(1)}$	requirement in Theorem 1
	$\frac{\log q}{\log 2\beta} \leq (m_1 + m_2)$	requirement in Theorem 2
	$q > 2\varepsilon\beta m_2 n^{1.5} \log n$	requirement in Theorem 2
γ	$\gamma = 8\varepsilon^2 \beta m_2 n \log^2 n$	approximating factor in Theorem 2
ε	$\varepsilon = 3$	$\mathrm{EF}(X^n + 1, 3) \leq 3$

– **Gen**(1^n): On input a security parameter n, the algorithm generates parameters as in Table 1. It defines $R_q = \mathbb{Z}_q[X]/\langle X^n + 1\rangle$, and independently picks two vectors $\mathbf{h} \leftarrow R_q^{m_1}$, $\mathbf{g} \leftarrow R_q^{m_2}$. Let $\mathcal{Q} = \{a \in R_q : \|a\|_\infty \leq \beta\}$. The message space is defined by $\mathcal{Q}^{m_1}$ and the randomness space is by $\mathcal{Q}^{m_2}$. This algorithm then publishes $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ as the public commitment key.
– **Com**$_{ck}(\mathbf{b}, \mathbf{r})$: On input a message $\mathbf{b} \in \mathcal{Q}^{m_1}$, and a randomly sampled randomness $\mathbf{r} \leftarrow \chi$, where χ is a distribution over $\mathcal{Q}^{m_2}$, this algorithm computes and outputs a commitment $c = \langle \mathbf{h}, \mathbf{b}\rangle + \langle \mathbf{g}, \mathbf{r}\rangle$, where the algebraic computations are done in R_q and the resulting commitment c can later be opened by unveiling the short $\mathbf{b}$ and $\mathbf{r}$.

A commitment scheme is said to be hiding, only if it reveals nothing about the committed value. The strongly binding property ensures that a sender cannot open the commitment to two different value-randomness pairs.

Theorem 3. *$\mathcal{CMT}$ with parameters in Table 1 is statistically hiding, if randomness is chosen from $[-\mathcal{B}, \mathcal{B}]^{m_2 n}$ uniformly at random, or if randomness is chosen according to $D_\sigma^{m_2 n}$. $\mathcal{CMT}$ is computationally biding, if f-SPP$_\gamma(\mathcal{I})$ is intractable to solve in the worst-case, where $f = X^n + 1$, and $\gamma = 8\varepsilon^2 \beta m_2 n \log^2 n$ is a polynomial in n.*

Proof. We start with the statistically hiding property. Focus on $\langle \mathbf{g}, \mathbf{r}\rangle$. It is an instance $f_{\mathbf{g}} \leftarrow \mathcal{K}(R_q, D, m_2)$. If $D = [-\mathcal{B}, \mathcal{B}]^{m_2 n}$, by Theorem 1 and the parameter settings, $\{(\mathbf{g}, f_{\mathbf{g}}(\mathbf{r})) : \mathbf{g} \leftarrow R_q^{m_2 n}, \mathbf{r} \leftarrow D\}$ is within statistical distance $\epsilon = \mathrm{negl}(n)$ to the uniform distribution over $R_q^{m_2 n} \times R_q$. Denote $\mathbf{g}$ by $(g_1, \ldots, g_{m_2}) \in R_q^{m_2}$ and let $\mathbf{G} = (Rot(g_1), \ldots, Rot(g_{m_2}))$. The foregoing discussion also implies that for all but at most $\mathrm{negl}(n)$ fraction of $\mathbf{g} \in R_q^{m_2}$, the columns of $\mathbf{G}$ generates $\mathbb{Z}_q^n$. Thus, by Lemma 5, there is a negligible function $\epsilon(m_2 n)$ such that $\eta_\epsilon(\Lambda^\perp(\mathbf{G})) \leq \omega(\log(m_2 n))$. If $\mathbf{r} \leftarrow D_\sigma^{m_2 n}$, and $\sigma \geq \omega(\sqrt{\log(m_2 n)})$, relying on Lemma 4, $\langle \mathbf{g}, \mathbf{r}\rangle$ is within statistical distance $2\epsilon(m_2 n) = \mathrm{negl}(n)$ of uniform distribution over $\mathbb{Z}_q^n$ (*i.e.*, R_q).

We proceed to consider the computationally strongly binding property. First notice that since $\sigma \geq \omega(\sqrt{\log(m_2 n)})$, then by Lemma 6, and Lemma 7, we have $\|\mathbf{r}\|_\infty \leq \beta$ with overwhelming probability, even if $\mathbf{r} \leftarrow D_\sigma^{m_2 n}$. Let $m = m_1 + m_2$. We observe that a pair of distinct openings to a commitment $c = \langle \mathbf{h}, \mathbf{b}\rangle + \langle \mathbf{g}, \mathbf{r}\rangle$ is also a solution of the collision problem $\mathrm{Col}_{\mathcal{K}}(h_{(\mathbf{h}\|\mathbf{g})})$ with respect to the generalized knapsack function family $\mathcal{K}(R_q, \mathcal{Q}, m)$ Moreover, with the parameter settings and according to Theorem 2, there is a polynomial time reduction from f-SPP$_\gamma(\mathcal{I})$ to $\mathrm{Col}_{\mathcal{K}}(h_{(\mathbf{h}\|\mathbf{g})})$. □

4.2 Sigma-Protocol for Commitment to 0 or 1

The Sigma-protocol designed in this section is named by $\Sigma_1 = (\mathcal{G}, \mathcal{P}, \mathcal{V})$, where $\mathcal{G}$ on input a security parameter n, generates the public parameters as in Table 2. $\mathcal{G}$ then picks $\mathbf{h} \leftarrow R_q^{m_1}$, $\mathbf{g} \leftarrow R_q^{m_2}$, so that $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ becomes the

Table 2. Parameter settings for Σ_1

Symbol	Setting	Explanation
n	$n = 2^k,\ k \in \mathbb{Z}^+$	as in $\mathcal{CMT}$
m_1	$m_1 = 1$	as in $\mathcal{CMT}$
m_2	$m_2 = \omega(1)$	as in $\mathcal{CMT}$
$\mathcal{B}$	$2\mathcal{B}+1 = n^{\Omega(1)}$	as in $\mathcal{CMT}$
σ	$\sigma = \omega(\sqrt{\log n})$	as in $\mathcal{CMT}$. Moreover, these parameters should satisfy the requirements of Lemma 8. $\alpha = 12$, and compute norm of the center T. Let the Gaussian parameter be αT
	$\sigma \geq 12$	
σ'	$\sigma' = \omega(\sqrt{\log(m_2 n)})$	
	$\sigma' = 12\mathcal{B}\sqrt{m_2 n}$	
σ''	$\sigma'' = \omega(\sqrt{\log(m_2 n)})$	
	$\sigma'' = (12 + 24\sigma\sqrt{n})\mathcal{B}n\sqrt{m_2}$	
M	$M = e^{289/288}$	$\alpha = 12$ in Lemma 8
β	$\beta \geq 4\sigma''\sqrt{m_2 n}$	as in $\mathcal{CMT}$ and $\sigma'' > \mathcal{B}$
q	$q = 5 \pmod 8$	prime in Lemma 1
	$q = n^{O(1)}$	as in $\mathcal{CMT}$
	$q > 4\sigma^2 n$	$\|z\|^2 \leq q$ in Lemma 1
	$\frac{\log q}{\log 2\beta} \leq (m_1 + m_2)$	as in $\mathcal{CMT}$
	$q > 6\beta m_2 n^{1.5}\log n$	as in $\mathcal{CMT}$ and $\varepsilon = 3$
ϵ	$\epsilon = 2^{-100}$	$\alpha = 12$ in Lemma 8
γ	$\gamma = 72\beta m_2 n \log^2 n$	as in $\mathcal{CMT}$ and $\varepsilon = 3$
T	$T = \mathcal{B}$	parameter in $\mathcal{R}(T)$
T'	$T' = 2n\sigma'\sqrt{m_2}$	parameter in $\mathcal{R}'(T')$

commitment key of $\mathcal{CMT}$ in Sect. 4.1. Depending on the common reference string (commitment key) generated by $\mathcal{G}$, protocol Σ_1 is for the following relations.

$$\mathcal{R}(T) = \{(ck, c, (b, \mathbf{r})) : b \in \{0,1\} \wedge \|\mathbf{r}\|_\infty \leq T \wedge c = \mathbf{Com}(b, \mathbf{r})\} \quad ,$$

$$\mathcal{R}'(T') = \{(ck, 2c, (2b, 2\cdot\mathbf{r})) : b \in \{0,1\} \wedge \|2\mathbf{r}\|_\infty \leq T' \wedge 2c = \mathbf{Com}(2b, 2\cdot\mathbf{r})\} \quad .$$

The details of the interactions between $\mathcal{P}$ and $\mathcal{V}$ are as follows. All the algebraic operations are done in R_q.

Algorithm $\mathcal{P}(ck, c, (b, \mathbf{r}))$:

– Initial message:
- Sample $a \leftarrow D_\sigma^n$, $\mathbf{s} \leftarrow D_{\sigma'}^{m_2 n}$, $\mathbf{t} \leftarrow D_{\sigma''}^{m_2 n}$.
- Compute $d = \mathbf{Com}(a, \mathbf{s})$.
- Compute $e = \mathbf{Com}(ab, \mathbf{t})$.
- Send $\mathrm{Cmt} = (d, e)$ to $\mathcal{V}$.

Algorithm $\mathcal{V}(ck, c, \mathrm{Cmt})$:

– Challenge:

- Send to $\mathcal{P}$ a challenge $x = X^w$, where $w \leftarrow \{0, \ldots, 2n-1\}$.

Algorithm $\mathcal{P}(ck, c, (b, \mathbf{r}), x)$:

– Response:

- Compute $f = x \cdot b + a$.
- Compute $\mathbf{y} = x \cdot \mathbf{r} + \mathbf{s}$.
- Compute $\mathbf{z} = (x - f) \cdot \mathbf{r} + \mathbf{t}$.
- Abort with probability

$$1 - \frac{D_\sigma(f)}{MD_{x\cdot b,\sigma}(f)} \cdot \frac{D^{m_2 n}_{\sigma'}(\mathbf{y})}{MD^{m_2 n}_{x\cdot\mathbf{r},\sigma'}(\mathbf{y})} \cdot \frac{D^{m_2 n}_{\sigma''}(\mathbf{z})}{MD^{m_2 n}_{(x-f)\cdot\mathbf{r},\sigma''}(\mathbf{z})} \ .$$

- Send $\mathrm{Rsp} = (f, \mathbf{y}, \mathbf{z})$ to $\mathcal{V}$.

Algorithm $\mathcal{V}(ck, c, \mathrm{Cmt}, x, \mathrm{Rsp})$:

– Verification:

Check if

- $\|f\| \leq \sigma\sqrt{n}$,
- $\|\mathbf{y}\| \leq \sigma'\sqrt{m_2 n}$,
- $\|\mathbf{z}\| \leq \sigma''\sqrt{m_2 n}$,
- $x \cdot c + d = \mathbf{Com}(f, \mathbf{y})$,
- $(x - f) \cdot c + e = \mathbf{Com}(0, \mathbf{z})$.

Output 1 if all the conditions hold, and output 0 otherwise.

Theorem 4. *Σ_1 is a Sigma-protocol for the relation $\mathcal{R}(T)$ and $\mathcal{R}'(T')$ with completeness error $1 - \left(\frac{1-\epsilon}{M}\right)^3$. It is 2-special sound if $\mathcal{CMT}$ with respect to $ck = (n, m_1, m_2, \beta, \mathbf{f}, \mathbf{g}, q)$ is computationally hiding. It is SHVZK if the $\mathcal{CMT}$ is statistically hiding.*

Proof. **Completeness:** With rejection samplings (Lemma 8), the distribution of f is statistically close to D_σ/M within statistical distance ϵ/M. Since $\sigma \geq \omega(\sqrt{\log n})$, according to Lemma 6 and Lemma 7, the l_2 norm of f is upper-bounded by $\sigma\sqrt{n}$ with probability $1 - \mathrm{negl}(n) - \epsilon/M$. If we consider $\epsilon = 2^{-100}$ as a negligible value, the above probability is overwhelming (otherwise, parameters in rejection samplings could be adjusted to make ϵ to be a negligible function in n, but we do not make such a discussion in this paper). The similar reasons lead us to the fact that $\|\mathbf{y}\| \leq \sigma'\sqrt{m_2 n}$, $\|\mathbf{z}\| \leq \sigma''\sqrt{m_2 n}$ with overwhelming probability. Next, through a simple deduction, we have

$$\begin{aligned} x \cdot c + d &= x \cdot \mathbf{Com}(b, \mathbf{r}) + \mathbf{Com}(a, \mathbf{s}) \\ &= \mathbf{Com}(x \cdot b + a, x \cdot \mathbf{r} + \mathbf{s}) \\ &= \mathbf{Com}(f, \mathbf{y}) \ , \end{aligned}$$

and if $b \in \{0, 1\}$,

$$\begin{aligned} (x - f) \cdot c + e &= (x(1 - b) - a) \cdot \mathbf{Com}(b, \mathbf{r}) + \mathbf{Com}(ab, \mathbf{t}) \\ &= \mathbf{Com}(b(1 - b)x - ba, (x - f) \cdot \mathbf{r}) + \mathbf{Com}(ab, \mathbf{t}) \\ &= \mathbf{Com}(0, (x - f)\mathbf{r} + \mathbf{t}) \\ &= \mathbf{Com}(0, \mathbf{z}) \ . \end{aligned}$$

Consequently, if $\mathcal{P}$ does not abort in the "Response" step, its response is able to pass the "Verification" step. According to Lemma 8, the probability that $\mathcal{P}$ outputs something is at least $\left(\frac{1-\epsilon}{M}\right)^3$, the completeness error is $1-\left(\frac{1-\epsilon}{M}\right)^3$.

2-special soundness: Let $(\mathrm{Cmt}, x, \mathrm{Rsp})$, $(\mathrm{Cmt}, x', \mathrm{Rsp}')$ be two accepting transcripts. We design a PPT extractor on input these transcripts acts as follows. From the first equation in the "Verification" step, we have

$$x \cdot c + d = \mathbf{Com}(f, \mathbf{y}), \;\; x' \cdot c + d = \mathbf{Com}(f', \mathbf{y}') \;\; .$$

By subtracting the former using the latter, we obtain

$$(x - x') \cdot c = \mathbf{Com}(f - f', \mathbf{y} - \mathbf{y}') \;\; .$$

By multiplying both sides by $2(x-x')^{-1}$, we have

$$\begin{aligned} 2c &= \mathbf{Com}(2(f-f')(x-x')^{-1}, (\mathbf{y}-\mathbf{y}') \cdot 2(x-x')^{-1}) \\ &= \mathbf{Com}(2\widehat{b}, 2 \cdot \widehat{\mathbf{r}}) \;\; , \end{aligned}$$

where $2\widehat{b} \stackrel{\text{def}}{=} 2(f-f')(x-x')^{-1}$, $2 \cdot \widehat{\mathbf{r}} \stackrel{\text{def}}{=} (\mathbf{y}-\mathbf{y}') \cdot 2(x-x')^{-1}$ are defined as a candidate of openings to $2c$. Since $q > 2$ is a prime, we have $\widehat{b} = \frac{f-f'}{x-x'}$. With Lemma 2, we observe that

$$\begin{aligned} \|2 \cdot \widehat{\mathbf{r}}\|_\infty = \|2(\mathbf{y}-\mathbf{y}')(x-x')^{-1}\|_\infty &\leq \|(\mathbf{y}-\mathbf{y}')\| \cdot \|2(x-x')^{-1}\| \\ &\leq \sqrt{n} \cdot (\|\mathbf{y}\| + \|\mathbf{y}'\|) \leq 2n\sigma'\sqrt{m_2} \leq T' \leq \beta \;\; , \end{aligned}$$

which shows that $2 \cdot \widehat{\mathbf{r}}$ is a valid randomness for $2c$ as described in the relation $\mathcal{R}'(T')$.

We proceed to explain the validity of $2\widehat{b}$. By the last equation in the verification step, we have

$$(x-f)2c + 2e = \mathbf{Com}(0, 2\mathbf{z}) \tag{1}$$
$$(x'-f')2c + 2e = \mathbf{Com}(0, 2\mathbf{z}') \tag{2}$$

Subtracting (1) with (2), we have

$$\begin{aligned} \widehat{c} &\stackrel{\text{def}}{=} (x - f - x' + f')2c \\ &= ((x-x') - (f-f')) \cdot \mathbf{Com}(2\widehat{b}, 2 \cdot \widehat{\mathbf{r}}) \\ &= \mathbf{Com}(2(f-f') - 2(f-f')^2(x-x')^{-1}, \\ &\quad 2 \cdot (\mathbf{y}-\mathbf{y}') - (\mathbf{y}-\mathbf{y}') \cdot 2(x-x')^{-1}(f-f')) \\ &= \mathbf{Com}(2(f-f')(1-(f-f')(x-x')^{-1}), \\ &\quad 2 \cdot (\mathbf{y}-\mathbf{y}') - (\mathbf{y}-\mathbf{y}') \cdot 2(x-x')^{-1}(f-f')) \\ &= \mathbf{Com}(0, 2 \cdot (\mathbf{z}-\mathbf{z}')) \end{aligned}$$

We observe the following facts on norms.

$$
\begin{aligned}
\|f - f'\| &\leq \|f\| + \|f'\| \leq 2\sigma\sqrt{n} \ . \\
\|v\|_\infty &\stackrel{\text{def}}{=} \|2(f-f')(1-(f-f')(x-x')^{-1})\|_\infty \\
&\leq \|f-f'\| \cdot \|2 - 2(f-f')(x-x')^{-1}\| \\
&\leq 2\sigma\sqrt{n}\left(2 + \sqrt{n} \cdot \|f-f'\| \cdot \|2(x-x')^{-1}\|\right) \\
&\leq 2\sigma\sqrt{n}\left(2 + \sqrt{n} \cdot 2\sigma\sqrt{n} \cdot \sqrt{n}\right) \\
&= \left(1 + \sigma n\sqrt{n}\right) 4\sigma\sqrt{n} \leq \beta \ . \\
\|\mathbf{s}\|_\infty &\stackrel{\text{def}}{=} \|2 \cdot (\mathbf{y} - \mathbf{y}') - (\mathbf{y} - \mathbf{y}') \cdot 2(x-x')^{-1}(f-f')\|_\infty \\
&\leq \|2 \cdot (\mathbf{y} - \mathbf{y}')\|_\infty + \|(\mathbf{y} - \mathbf{y}') \cdot 2(x-x')^{-1}(f-f')\|_\infty \\
&\leq 2 \cdot \|\mathbf{y} - \mathbf{y}'\| + \sqrt{n} \cdot \|\mathbf{y} - \mathbf{y}'\| \cdot \|2(x-x')^{-1}\| \cdot \|f - f'\| \\
&\leq 4\sigma'\sqrt{m_2 n} + \sqrt{n} \cdot 2\sigma'\sqrt{m_2 n} \cdot \sqrt{n} \cdot 2\sigma\sqrt{n} \\
&= (1 + \sigma n\sqrt{n}) 4\sigma'\sqrt{m_2 n} \leq \beta \ . \\
\|2 \cdot (\mathbf{z} - \mathbf{z}')\|_\infty &\leq 2 \cdot (\|\mathbf{z}\| + \|\mathbf{z}'\|) = 4\sigma''\sqrt{m_2 n} \leq \beta \ .
\end{aligned}
$$

By the computationally binding assumption (otherwise, $(v, \mathbf{s})$ and $(0, 2 \cdot (\mathbf{z} - \mathbf{z}'))$ are a pair of distinct openings to the commitment $\widehat{c}$), it is with overwhelming probability that

$$v = 2(f-f')(1-(f-f')(x-x')^{-1}) = 0 \ .$$

Since $\|f - f'\|^2 \leq 4\sigma^2 n < q$, it is invertible in R_q by Lemma 1. As a result, we have

$$f - f' = 0, \text{ or } 1 - (f-f')(x-x')^{-1} = 0$$

In the former case, we have $\widehat{b} = \frac{f-f'}{x-x'} = 0$, and in the latter case, we have $\widehat{b} = \frac{f-f'}{x-x'} = 1$. Finally, $\left(2\widehat{b}, 2 \cdot \widehat{\mathbf{r}}\right)$ is a valid witness for $2c$ in $\mathcal{R}'(T')$.

SHVZK: On input a challenge x, the simulator aborts with probability $1 - \left(\frac{1-\epsilon}{M}\right)^3$. Otherwise, it samples $f \leftarrow D_\sigma^n, \mathbf{y} \leftarrow D_{\sigma'}^{m_2 n}, \mathbf{z} \leftarrow D_{\sigma''}^{m_2 n}$. The distributions of these simulated responses are statistically close to that in a real proof. It then computes $d = \mathbf{Com}(f, \mathbf{y}) - x \cdot c$, $e = \mathbf{Com}(0, \mathbf{z}) - (x - f) \cdot c$. Since d, e, are uniquely determined by the equations in the verification step and the responses, the resulting distributions of d and e are identical to that in a real proof, thus resulting the SHVZK property. □

4.3 Sigma-Protocol for One-Out-of-μ Commitments

The Sigma-protocol designed in this section is named by $\Sigma_2 = (\mathcal{G}, \mathcal{P}, \mathcal{V})$, where $\mathcal{G}$ on input a security parameter n and the maximum number of commitments μ, generates the public parameters as in Table 3. $\mathcal{G}$ then picks $\mathbf{h} \leftarrow R_q^{m_1}$, $\mathbf{g} \leftarrow R_q^{m_2}$, so that $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ becomes the commitment key of $\mathcal{CMT}$ in Sect.

Table 3. Parameter settings for Σ_2

Symbol	Setting	Explanation
n	$n = 2^k,\ k \in \mathbb{Z}^+$	as in $\mathcal{CMT}$
m_1	$m_1 = 1$	as in $\mathcal{CMT}$
m_2	$m_2 = \omega(1)$	as in $\mathcal{CMT}$
$\mathcal{B}$	$2\mathcal{B}+1 = n^{\Omega(1)}$	as in $\mathcal{CMT}$
s	$s = \sqrt{\tau}\sigma$	Rejection sampling τ
s'	$s' = \sqrt{\tau+1}\sigma'$	(resp., $\tau+1$, τ) vectors
s''	$s'' = \sqrt{\tau}\sigma''$	of Σ_1 simultaneously
M	$M = e^{289/288}$	as in Σ_1
β	$\beta \geq (\tau+1)\cdot n^{\tau-0.5}\cdot s'\sqrt{m_2 n}$	as in $\mathcal{CMT}$ and $s' > \mathcal{B}$
q	$q = 5 \pmod 8$	as in Σ_1
	$q = n^{O(1)}$	as in $\mathcal{CMT}$
	$q > 4s^2 n$	as in Σ_1 but with s
	$\frac{\log q}{\log 2\beta} \leq (m_1+m_2)$	as in $\mathcal{CMT}$
	$q > 6\beta m_2 n^{1.5}\log n$	as in $\mathcal{CMT}$
ϵ	$\epsilon = 2^{-100}$	as in Σ_1
γ	$\gamma = 72\beta m_2 n\log^2 n$	as in Σ_1
μ	$\mu = 2^i, i \in \mathbb{Z}^+$	number of participants
τ	$\tau = \log\mu$	binary length of μ
T	$T = \mathcal{B}$	parameter in $\mathcal{R}(T)$
T'	$T' = (\tau+1)\cdot n^{\tau-0.5}\cdot s'\sqrt{m_2 n}$	parameter in $\mathcal{R}'(T')$

4.1. Depending on the common reference string (commitment key) generated by $\mathcal{G}$, protocol Σ_2 is for the following relations.

$$\mathcal{R}(T) = \left\{ (ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), (\ell, \mathbf{w})) : \begin{array}{c} \forall i \in [0,\mu), u_i \in R_q \wedge \\ \|\mathbf{w}\|_\infty \leq T \wedge \ell \in [0,\mu) \wedge \\ u_\ell = \mathbf{Com}(0, \mathbf{w}) \wedge \\ \mathbf{g}' \in R_q^{m_2} \wedge \eta = \langle \mathbf{g}', \mathbf{w} \rangle \end{array} \right\},$$

$$\mathcal{R}'(T') = \left\{ (ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), (\ell, \mathbf{w}')) : \begin{array}{c} \forall i \in [0,\mu), u_i \in R_q \wedge \\ \|\mathbf{w}'\|_\infty \leq T' \wedge \ell \in [0,\mu) \wedge \\ 2^\tau u_\ell = \mathbf{Com}(0, \mathbf{w}') \wedge \\ \mathbf{g}' \in R_q^{m_2} \wedge 2^\tau \eta = \langle \mathbf{g}', \mathbf{w}' \rangle \end{array} \right\}.$$

To construct a logarithmic size ring signature, Groth and Kohlweiss proposed a technique to compute the coefficients of a polynomial in the indeterminate x over finite field $\mathbb{Z}_q^*$ in advance, where x is a hash value computed later [14]. Our protocol also requires such a computation.

Let integer ℓ be in $[0, \mu-1]$. In the scheme, we will see that $f_j = \ell_j \cdot x + a_j$. Define $f_{j,1} = f_j = \ell_j \cdot x + a_j = \delta_{1\ell_j} x + a_j$ and $f_{j,0} = x - f_j = (1-\ell_j)\cdot x - a_j =$

$\delta_{0\ell_j} x - a_j$. Then for each $i \in [0, \mu-1]$, the product $\prod_{j=1}^{\tau} f_{j,i_j}$ is a polynomial in x of the form

$$p_i(x) = \prod_{j=1}^{\tau} (\delta_{i_j \ell_j} x) + \sum_{k=0}^{\tau-1} p_{i,k} x^k = \delta_{i\ell} x^\tau + \sum_{k=0}^{\tau-1} p_{i,k} x^k \quad , \tag{3}$$

where $p_{i,k}$ is the coefficient of the k-th degree term, and can be efficiently computed if $(a_j)_{j=1}^{\tau}$, i and ℓ are given.

We proceed to introduce the details of the interactions between $\mathcal{P}$ and $\mathcal{V}$. All the algebraic operations are done in R_q.

Algorithm $\mathcal{P}(ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), (\ell, \mathbf{w}))$:

– Initial messages:
- Parse ℓ as its binary expression $\ell = (\ell_1, \ldots, \ell_\tau)$
- For j from 1 to τ
  - Sample $\mathbf{r}_j \leftarrow [-\mathcal{B}, \mathcal{B}]^{m_2 n}$, $a_j \leftarrow D_s^n$, $\mathbf{s}_j \leftarrow D_{s'}^{m_2 n}$, $\mathbf{t}_j \leftarrow D_{s''}^{m_2 n}$
  - Compute $c_j = \mathbf{Com}(\ell_j, \mathbf{r}_j)$
  - Compute $d_j = \mathbf{Com}(a_j, \mathbf{s}_j)$
  - Compute $e_j = \mathbf{Com}(a_j \ell_j, \mathbf{t}_j)$
- For k from 0 to $\tau - 1$
  - Sample $\rho_k \leftarrow D_{s'\sqrt{\tau^{-1}}}^{m_2 n}$
  - Compute $v_k = (\sum_{i=0}^{\mu-1} u_i p_{i,k}) + \mathbf{Com}(0, \rho_k)$, where $p_{i,k}$ is computed as in (3)
  - Compute $v'_k = \langle \mathbf{g}', \rho_k \rangle$
- Send $\mathrm{Cmt} = \left(c_j, d_j, e_j, v_{j-1}, v'_{j-1}\right)_{j=1}^{\tau}$ to $\mathcal{V}$

Algorithm $\mathcal{V}(ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), \mathrm{Cmt})$:

– Challenge:
- Send to $\mathcal{P}$ a challenge $x = X^w$, where $w \leftarrow \{0, \ldots, 2n-1\}$

Algorithm $\mathcal{P}(ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), (\ell, \mathbf{w}), x)$:

– Responses:
- For j from 1 to τ, compute
  - $f_j = \ell_j \cdot x + a_j$
  - $\mathbf{y}_j = x \cdot \mathbf{r}_j + \mathbf{s}_j$
  - $\mathbf{z}_j = (x - f_j) \cdot \mathbf{r}_j + \mathbf{t}_j$
- Compute $\mathbf{v} = x^\tau \cdot \mathbf{w} - \sum_{k=0}^{\tau-1} x^k \cdot \rho_k$
- Denoting $\mathbf{f} = (f_1 \| \ldots \| f_\tau)$, $\mathbf{y} = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_\tau \| \mathbf{v})$, $\mathbf{z} = (\mathbf{z}_1 \| \ldots \| \mathbf{z}_\tau)$, $\mathbf{c} = x \cdot (\ell_1, \ldots, \ell_\tau)$, $\mathbf{c}' = (x \cdot \mathbf{r}_1 \| \ldots, \| x \cdot \mathbf{r}_\tau \| x^\tau \cdot \mathbf{w})$, $\mathbf{c}'' = ((x - f_1) \cdot \mathbf{r}_1 \| \ldots \| (x - f_\tau) \cdot \mathbf{r}_\tau)$, $\mathcal{P}$ aborts with probability

$$1 - \frac{D_s^{\tau n}(\mathbf{f})}{M D_{\mathbf{c},s}^{\tau n}(\mathbf{f})} \cdot \frac{D_{s'}^{(\tau+1)m_2 n}(\mathbf{y})}{M D_{\mathbf{c}',s'}^{(\tau+1)m_2 n}(\mathbf{y})} \cdot \frac{D_{s''}^{\tau m_2 n}(\mathbf{z})}{M D_{\mathbf{c}'',s''}^{\tau m_2 n}(\mathbf{z})}$$

- Send $\mathrm{Rsp} = (\mathbf{f}, \mathbf{y}, \mathbf{z})$ to $\mathcal{V}$.

Algorithm $\mathcal{V}(ck, (u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta), \mathrm{Cmt}, x, \mathrm{Rsp})$:

– Verification

- For j from 1 to τ, check if
  - $\|f_j\| \leq s\sqrt{n}$
  - $\|\mathbf{y}_j\| \leq s'\sqrt{m_2 n}$
  - $\|\mathbf{z}_j\| \leq s''\sqrt{m_2 n}$
  - $x \cdot c_j + d_j = \mathbf{Com}(f_j, \mathbf{y}_j)$
  - $(x - f_j) \cdot c_j + e_j = \mathbf{Com}(0, \mathbf{z}_j)$
- Check if $\|\mathbf{v}\| \leq s'\sqrt{m_2 n}$
- Check if $\eta x^\tau + \sum_{k=0}^{\tau-1} v'_k(-x^k) = \langle \mathbf{g}', \mathbf{v} \rangle$
- Check if $\sum_{i=0}^{\mu-1}(u_i \prod_{j=1}^{\tau} f_{j,i_j}) + \sum_{k=0}^{\tau-1} v_k(-x^k) = \mathbf{Com}(0, \mathbf{v})$

If all above conditions are satisfied, output 1 to accept; otherwise, output 0 to reject.

Theorem 5. *Σ_2 with parameters in Table 3 is a Sigma-protocol for the relation $\mathcal{R}(T)$ and $\mathcal{R}'(T')$ with completeness error $1 - \left(\frac{1-\epsilon}{M}\right)^3$. It is $(\tau+1)$-special sound if $\mathcal{CMT}$ is computationally binding. It is SHVZK if the commitment scheme $\mathcal{CMT}$ is statistically hiding.*

Proof. **Completeness:** The validity regarding to f_j, $\mathbf{y}_j$, and $\mathbf{z}_j$ have been shown in the proof of Σ_1, except that the Gaussian parameters were chosen to support Σ_2. Thus, we focus on the new parameters and equations emerged in Σ_2. For $k \in [0, \tau)$, $\rho_k \leftarrow D^{m_2 n}_{s'\sqrt{\tau^{-1}}}$. Since $\|x^k \rho_k\| = \|\rho_k\|$, $\sum_{k=0}^{\tau-1} x^k \rho_k$ follows the distribution $D^{m_2 n}_{s'}$. With rejection samplings, the distribution of $\mathbf{y} = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_\tau \| \mathbf{v})$ is close to $D^{(\tau+1)m_2 n}_{s'}/M$ within statistical distance ϵ/M. Since $s' = \sqrt{\tau+1} \cdot \sigma' \geq \sqrt{\tau+1} \cdot \omega(\sqrt{\log(m_2 n)}) \geq \omega(\sqrt{\log(m_2 n)})$, according to Lemma 6 and Lemma 7, the l_2 norm of $\mathbf{v}$ is upper-bounded by $s'\sqrt{m_2 n}$ with overwhelming probability.

Later, we observe that if a transcript is generated honestly, then

$$\begin{aligned}\eta x^\tau + \sum_{k=0}^{\tau-1} v'_k(-x^k) &= \langle \mathbf{g}', \mathbf{w}_\ell \rangle x^\tau - \sum_{k=0}^{\tau-1} \langle \mathbf{g}', \rho_k \rangle (x^k) \\ &= \langle \mathbf{g}', x^\tau \cdot \mathbf{w}_\ell - \sum_{k=0}^{\tau-1} x^k \cdot \rho_k \rangle = \langle \mathbf{g}', \mathbf{v} \rangle \ ,\end{aligned}$$

and

$$
\begin{aligned}
&\sum_{i=0}^{\mu-1}(u_i\prod_{j=1}^{\tau}f_{j,i_j})+\sum_{k=0}^{\tau-1}v_k(-x^k)\\
&=\sum_{i=0}^{\mu-1}u_i\left(\delta_{i\ell}x^\tau+\sum_{k=0}^{\tau-1}p_{i,k}x^k\right)+\sum_{k=0}^{\tau-1}\left(\sum_{i=0}^{\mu-1}u_ip_{i,k}+\mathbf{Com}(0,\rho_k)\right)(-x^k)\\
&=u_\ell x^\tau+\sum_{i=0}^{\mu-1}\sum_{k=0}^{\tau-1}\left(u_ip_{i,k}x^k-u_ip_{i,k}x^k\right)-\sum_{k=0}^{\tau-1}\mathbf{Com}(0,\rho_k)x^k\\
&=\mathbf{Com}(0,x^\tau\cdot\mathbf{w}-\sum_{k=0}^{\tau-1}\rho_kx^k)\\
&=\mathbf{Com}(0,\mathbf{v})\ .
\end{aligned}
$$

Consequently, if $\mathcal{P}$ does not abort in the "Response" step, its response is able to pass the "Verification" step. According to Lemma 8, the probability that $\mathcal{P}$ outputs something is at least $\left(\frac{1-\epsilon}{M}\right)^3$. The completeness error is $1-\left(\frac{1-\epsilon}{M}\right)^3$.

$(\tau+1)$-special soundness: Let Cmt be an initial message, and $(x_i,\mathrm{Rsp}^{(\theta)})_{\theta=0}^{\tau}$ be $\tau+1$ distinct accepting challenge-response pairs on Cmt. The PPT extractor designed in this section acts as follows.

By the 2-special soundness shown in Sect. 4.2, we obtain valid openings of $\{2c_j\}_{j=1}^{\tau}$. Denoting them by $\{(2\ell_j,2\cdot\mathbf{r}_j)\}_{j=1}^{\tau}$, respectively, they satisfy $\ell_j\in\{0,1\}$, $\|2\cdot\mathbf{r}_j\|_\infty\leq 2ns'\sqrt{m_2}$ with overwhelming probability. Denoting ℓ by $\ell=\sum_{j=1}^{\tau}\ell_j2^{j-1}$, then $\ell\in[0,\mu-1]$ satisfies the requirement of $R'(T')$.

Since the transcripts are accepting, by the first equation in the verification step, we know

$$
2d_j=\mathbf{Com}(2(f_j^{(\theta)}-x^{(\theta)}\ell_j),2\mathbf{y}_j^{(\theta)}-2x^{(\theta)}\mathbf{r}_j),\ \text{for all}\ \theta\in[0,\tau],j\in[1,\tau]\ ,
$$

and obtain the following upper-bound on norms.

$$
\begin{aligned}
&\|2(f_j^{(\theta)}-x^{(\theta)}\ell_j)\|_\infty\leq 2\cdot\|f_j^{(\theta)}\|_\infty+\|2x^{(\theta)}\ell_j\|_\infty\leq 2s\sqrt{n}+2\leq\beta\ ,\\
&\|2\mathbf{y}_j^{(\theta)}-2x^{(\theta)}\mathbf{r}_j\|_\infty\leq\|2\mathbf{y}_j^{(\theta)}\|+\|2x^{(\theta)}\mathbf{r}_j\|_\infty\leq 2s'\sqrt{m_2n}+2ns'\sqrt{m_2}\leq\beta\ .
\end{aligned}
$$

As a result

$$
2a_j^{(\theta)}=2(f_j^{(\theta)}-x^{(\theta)}\ell_j)\,,\ \ 2\mathbf{s}_j=2\mathbf{y}_j^{(\theta)}-2x^{(\theta)}\mathbf{r}_j\ ,
$$

are a pair of valid openings to $2d_j$.

Because of the computationally binding assumption on $\mathcal{CMT}$, it is with overwhelming probability that $2a_j^{(\theta)}=2a_j^{(\theta')}\overset{\text{def}}{=}2a_j$, $2\mathbf{s}_j^{(\theta)}=2\mathbf{s}_j^{(\theta')}\overset{\text{def}}{=}2\mathbf{s}_j$ for all $\theta,\theta'\in[0,\tau]$ (otherwise, there exists at least one pair of distinct openings of $2d_j$ within them). Consequently, for all $\theta\in[0,\tau]$, $j\in[1,\tau]$, $2f_j^{(\theta)}$ is of the form $2f_j^{(\theta)}=2\ell_jx^{(\theta)}+2a_j$, and could be viewed as a polynomial in the indeterminate

$x^{(\theta)}$. Subsequently, by the last equation in the verification step

$$\begin{aligned}
&\sum_{i=0}^{\mu-1}(u_i \prod_{j=1}^{\tau} 2f_{j,i_j}^{(\theta)}) + 2^\tau \sum_{k=0}^{\tau-1} v_k(-(x^{(\theta)})^k) \\
&= 2^\tau u_\ell (x^{(\theta)})^\tau + 2^\tau \sum_{i=0}^{\mu-1}\sum_{k=0}^{\tau-1} \left(u_i p_{i,k}(x^{(\theta)})^k\right) - 2^\tau \sum_{k=0}^{\tau-1} v_k (x^{(\theta)})^k \\
&= \mathbf{Com}(0, 2^\tau \mathbf{v}^{(\theta)}) \ ,
\end{aligned}$$

we observed that the second line of the above equation could be regarded as a polynomial in the indeterminate $x^{(\theta)}$ of degree τ, and the coefficient of the τ-th degree term is $2^\tau u_\ell$. As we have $\tau+1$ distinct accepting transcripts to the same initial messages, we have

$$\begin{pmatrix} 1 & (x^{(0)})^1 & \cdots & (x^{(0)})^\tau \\ 1 & (x^{(1)})^1 & \cdots & (x^{(1)})^\tau \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x^{(\tau)})^1 & \cdots & (x^{(\tau)})^\tau \end{pmatrix} \begin{pmatrix} \mathrm{coeff}_0 \\ \mathrm{coeff}_1 \\ \vdots \\ 2^\tau u_\ell \end{pmatrix} = 2^\tau \cdot \begin{pmatrix} \mathbf{Com}(0, \mathbf{v}^{(0)}) \\ \mathbf{Com}(0, \mathbf{v}^{(1)}) \\ \vdots \\ \mathbf{Com}(0, \mathbf{v}^{(\tau)}) \end{pmatrix} \ , \tag{4}$$

where for $k \in [0, \tau)$, coeff_k is the coefficient of the k-th degree term, but is not required to be computed explicitly. We then compute

$$\left(\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\tau)}\right) = \left(0, 0, \ldots, 0, 1\right) \begin{pmatrix} 1 & (x^{(0)})^1 & \cdots & (x^{(0)})^\tau \\ 1 & (x^{(1)})^1 & \cdots & (x^{(1)})^\tau \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x^{(\tau)})^1 & \cdots & (x^{(\tau)})^\tau \end{pmatrix}^{-1} \ .$$

Then, by left multiplying $(\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\tau)})$ to (4), we have

$$\begin{aligned}
&\left(\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\tau)}\right) \begin{pmatrix} 1 & (x^{(0)})^1 & \cdots & (x^{(0)})^\tau \\ 1 & (x^{(1)})^1 & \cdots & (x^{(1)})^\tau \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x^{(\tau)})^1 & \cdots & (x^{(\tau)})^\tau \end{pmatrix} \begin{pmatrix} \mathrm{coeff}_0 \\ \mathrm{coeff}_1 \\ \vdots \\ 2^\tau u_\ell \end{pmatrix} \\
&= \left(0, 0, \ldots, 0, 1\right) \begin{pmatrix} \mathrm{coeff}_0 \\ \mathrm{coeff}_1 \\ \vdots \\ 2^\tau u_\ell \end{pmatrix} = 2^\tau \cdot u_\ell \\
&= \mathbf{Com}\left(0, 2^\tau \sum_{\theta=0}^{\tau} \alpha^{(\theta)} \mathbf{v}^{(\theta)}\right) \\
&= \left(\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\tau)}\right) \begin{pmatrix} \mathbf{Com}(0, 2^\tau \mathbf{v}^{(0)}) \\ \mathbf{Com}(0, 2^\tau \mathbf{v}^{(1)}) \\ \vdots \\ \mathbf{Com}(0, 2^\tau \mathbf{v}^{(\tau)}) \end{pmatrix} \ .
\end{aligned}$$

Defining the candidate of extracted witness by $\mathbf{w}' = 2^\tau \sum_{\theta=0}^{\tau} \alpha^{(\theta)}\mathbf{v}^{(\theta)}$, we observe that

$$\begin{aligned}\|\mathbf{w}'\|_\infty = \left\| 2^\tau \sum_{\theta=0}^{\tau} \alpha^{(\theta)}\mathbf{v}^{(\theta)} \right\|_\infty &\leq (\tau+1)\cdot \max_{\theta\in[0,\tau]} \left\| 2^\tau \alpha^{(\theta)}\mathbf{v}^{(\theta)} \right\|_\infty \\ &\leq (\tau+1)\cdot \max_{\theta\in[0,\tau]} \|2^\tau\alpha^{(\theta)}\| \cdot \|\mathbf{v}^{(\theta)}\| \\ &\leq (\tau+1)\cdot n^{\tau-0.5}\cdot \|\mathbf{v}^{(\theta)}\| \\ &\leq (\tau+1)\cdot n^{\tau-0.5}\cdot s'\sqrt{m_2 n} \leq \beta \ ,\end{aligned}$$

where the second inequality is from Lemma 2, and the third inequality is depending on Lemma 3.

Similarly, from the equation corresponding to η in the verification step, we have

$$2^\tau \cdot \eta (x^{(\theta)})^\tau + 2^\tau \cdot \sum_{k=0}^{\tau-1} v'_k(-(x^{(\theta)})^k) = \langle \mathbf{g}', 2^\tau \cdot \mathbf{v}^{(\theta)} \rangle \ ,$$

so that with the $\tau+1$ accepting transcripts, we obtain

$$\begin{pmatrix} 1 & (x^{(0)})^1 & \cdots & (x^{(0)})^\tau \\ 1 & (x^{(1)})^1 & \cdots & (x^{(1)})^\tau \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x^{(\tau)})^1 & \cdots & (x^{(\tau)})^\tau \end{pmatrix} \begin{pmatrix} 2^\tau v'_0 \\ 2^\tau v'_1 \\ \vdots \\ 2^\tau \eta \end{pmatrix} = 2^\tau \cdot \begin{pmatrix} \langle \mathbf{g}', \mathbf{v}^{(0)} \rangle \\ \langle \mathbf{g}', \mathbf{v}^{(1)} \rangle \\ \vdots \\ \langle \mathbf{g}', \mathbf{v}^{(\tau)} \rangle \end{pmatrix} \ .$$

By left multiplying $(\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(\tau)})$, we observe that

$$2^\tau \eta = \langle \mathbf{g}', 2^\tau \sum_{\theta=0}^{\tau} \alpha^{(\theta)}\mathbf{v}^{(\theta)} \rangle = \langle \mathbf{g}', \mathbf{w}' \rangle \ .$$

Thus, $(\ell, \mathbf{w}')$ is a valid witness for the statement $(u_0, \ldots, u_{\mu-1}, \mathbf{g}', \eta)$ in the relation $\mathcal{R}'(T')$ introduced in Sect. 4.3.

SHVZK: On input a challenge x, the simulator with probability $1-((1-\epsilon)/M)^3$ aborts. Otherwise, for $j \in [1,\tau]$, it samples $f_j \leftarrow D_s^n$, $\mathbf{y}_j \leftarrow D_{s'}^{m_2 n}$, $\mathbf{z}_j \leftarrow D_{s''}^{m_2 n}$, and $\mathbf{v} \leftarrow D_{s'}^{m_2 n}$. The distributions of these simulated responses are statistically close to that in a real proof. Then it randomly picks $\ell \leftarrow [0,\mu)$. For $j \in [1,\tau]$, it picks $\mathbf{r}_j \leftarrow [-\mathcal{B}, \mathcal{B}]$ and computes $c_j = \mathbf{Com}(\ell_j, \mathbf{r}_j)$. $(c_j)_{j=1}^\tau$ are statistically indistinguishable from that of a real proof since $\mathcal{CMT}$ is statistically hiding.

Subsequently, for $j \in [1,\tau)$, let $a_j = f_k - \ell_j x$, and compute $(p_{i,k})_{i\in[0,\mu),k\in[1,\tau)}$ as in (3), by using $(a_j)_{j=1}^\tau$, x and $(\ell_j)_{j=1}^\tau$. For $k \in [1,\tau)$, it picks $\rho_k \leftarrow D_{s'\sqrt{\tau^{-1}}}^{m_2 n}$, and computes $v'_k = \langle \mathbf{g}', \rho_k \rangle$, $v_k = (\sum_{i=0}^{\mu-1} u_i p_{i,k}) + \mathbf{Com}(0, \rho_k)$. Consequently, for $k \in [1,\tau)$, v_k and v'_k are statistically uniformly distributed in R_q^n and they are pairwise dependent since they use the same randomness as in a real proof.

Since $(d_j)_{j=1}^{\tau}$, $(e_j)_{j=1}^{\tau}$, v_0, v_0' are uniquely determined by the corresponding verification equations and the generated parameters above, it computes

$$\begin{aligned}
d_j &= \mathbf{Com}(f_j, \mathbf{y}_j) - x \cdot c_j \ \ , \text{ for } j \in [1, \tau] \\
e_j &= \mathbf{Com}(0, \mathbf{z}_j) - (x - f_j) \cdot c_j \ \ , \text{ for } j \in [1, \tau] \\
v_0 &= \sum_{i=0}^{\mu-1} (u_i \prod_{j=1}^{\tau} f_{j,i_j}) + \sum_{k=1}^{\tau-1} v_k(-x^k) - \mathbf{Com}(0, \mathbf{v}) \\
v_0' &= \eta x^{\tau} + \sum_{k=1}^{\tau-1} v_k'(-x^k) - \langle \mathbf{g}', \mathbf{v} \rangle \ \ .
\end{aligned}$$

By the foregoing discussion, if the simulator does not abort, then the outputting transcript $(c_j, d_j, e_j, v_{j-1}, v_{j-1}')_{j=1}^{\tau}$, x, $((f_j)_{j=1}^{\tau}, (\mathbf{y}_j)_{j=1}^{\tau}, \mathbf{v}, (\mathbf{z}_j)_{j=1}^{\tau})$ is statistically indistinguishable from that of a real proof. As a result, Σ_2 is SHVZK. □

5 Linkable Ring Signature Based on Ideal-Lattices

In this section, we present a short ideal-lattice-based linkable ring signature as a counterpart of the ring signature in [14]. We notice that a classic edition of the current scheme can be built by instead using any cyclic group as long as its underlying DLP is hard. We propose a linkable ring signature based on the ECDLP, and discuss how to implement this signature with ECC in App. A. After reading it, one shall see that the designs of the ECDLP-based scheme are much more succinct than those of ideal-lattice-based one.

5.1 Parameters Settings

The parameter settings are in Table 4. The linkable ring signature in Sect. 5.2 is obtained by transforming the Sigma-protocol Σ_2 in Sect. 4.3 non-interactive, so that all the parameters except for r, r' are chosen to ensure the completeness and security of Σ_2. The statement that the signature scheme wishes to prove is $(vk_0, vk_1, \ldots, vk_{\mu-1}, \mathcal{H}_2(event), \eta)$. The goal is to convince the verifier that the signer knows one secret key of $vk_1, \ldots, vk_{\mu-1}$, and the linking tag η is generated with *event* and the secret key the signer holds. To reduce the soundness of Σ_2 to a negligible level, it is iterated by r times, as in [10]. However, this will increase the total completeness error, if rejection samplings in Σ_2 are done individually in each iteration. Instead, in the signature scheme, for example, the vectors $\mathbf{f}_1, \ldots, \mathbf{f}_r$ generated by all the iterations of Σ_2 are collected and are rejection sampled simultaneously, so that we could obtain a constant overall completeness error at $1 - \left(\frac{1-\epsilon}{M}\right)^3$ as in Σ_2. Nevertheless, by Lemma 8, since the upper-bounded norms of the vectors being rejection sampled are $\sqrt{r}$ times larger than that in Σ_2 (because of the dimension of the vectors), if we hope to achieve the above goal, we have to set the Gaussian parameters in the signature scheme $\sqrt{r}$ times larger than that in Σ_2. Other parameters corresponding to those Gaussian parameters are modified accordingly.

Table 4. Parameter settings for $\mathcal{LRS}$

Symbol	Setting	Explanation
n	$n = 2^k,\ k \in \mathbb{Z}^+$	as in $\mathcal{CMT}$
m_1	$m_1 = 1$	as in $\mathcal{CMT}$
m_2	$m_2 = \omega(1)$	as in $\mathcal{CMT}$
$\mathcal{B}$	$2\mathcal{B}+1 = n^{\Omega(1)}$	as in $\mathcal{CMT}$
s	$s = \sqrt{\tau r}\sigma$	Rejection sampling τr
s'	$s' = \sqrt{(\tau+1)r}\sigma'$	(resp., $(\tau+1)r$, τr) vectors
s''	$s'' = \sqrt{\tau r}\sigma''$	of Σ_1 simultaneously
M	$M = e^{289/288}$	as in Σ_1
β	$\beta \geq (\tau+1)\cdot n^{\tau-0.5}\cdot s'\sqrt{m_2 n}$	as in Σ_2
q	$q = 5 \pmod 8$	as in Σ_1
	$q = n^{O(1)}$	as in $\mathcal{CMT}$
	$q > 4s^2 n$	as in Σ_2
	$\frac{\log q}{\log 2\beta} \leq (m_1+m_2)$	as in $\mathcal{CMT}$
	$q > 6\beta m_2 n^{1.5}\log n$	as in $\mathcal{CMT}$
ϵ	$\epsilon = 2^{-100}$	as in Σ_1
γ	$\gamma = 72\beta m_2 n\log^2 n$	as in Σ_1
μ	$\mu = 2^i, i \in \mathbb{Z}^+$	as in Σ_2
τ	$\tau = \log\mu$	as in Σ_2
T	$T = \mathcal{B}$	as in Σ_2
T'	$T' = (\tau+1)\cdot n^{\tau-0.5}\cdot s'\sqrt{m_2 n}$	as in Σ_2
r	$r = n/(\log(2n)-\log\tau)$	reducing soundness error
r'	$r' = \frac{-n}{\log(1-1/M^2)}$	**Sign** always halts

5.2 Linkable Ring Signature Scheme

We proceed to introduce the linkable ring signature. It is consists of a tuple of PPT algorithms $\mathcal{LRS} = (\mathbf{Setup}, \mathbf{KGen}, \mathbf{Sign}, \mathbf{Vfy}, \mathbf{Link})$. The details of those algorithms are shown below.

- **Setup**(n, μ): On input μ and security parameter n, the algorithm initiates the system with the following rules
  - Generate $m_1, m_2, \mathcal{B}, s, s', s'', M, \beta, q, r, r'$ as in Table 4
  - Let $R_q = \mathbb{Z}_q[X]/\langle X^n+1\rangle$. All operations in this system are done in R_q
  - Randomly pick $\mathbf{h} \leftarrow R_q^{m_1}$, $\mathbf{g} \leftarrow R_q^{m_2}$
  - $ck \stackrel{\text{def}}{=} (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ becomes a commitment key of $\mathcal{CMT}$
  - Select two hash functions $\mathcal{H}_1 : \{0,1\}^* \to [0, 2n)^r$, $\mathcal{H}_2 : \{0,1\}^* \to R_q^{m_2}$

  All parameters generated by **Setup** are published as the global parameter pp, which is a default input of the other algorithms.
- **KGen**(pp): This algorithm randomly chooses $\mathbf{w} \in [-\mathcal{B}, \mathcal{B}]^{m_2 n}$ and parses it as a vector in $R_q^{m_2}$. Then it computes $u = \mathbf{Com}(0, \mathbf{w})$. The user's verifying key is $vk = u$ and the singing key is $sk = \mathbf{w}$.

– **Sign**$(sk_\ell, \mathrm{msg}, event, L)$: On input the participants $L = (u_i)_{i=0}^{\mu-1}$ and a message msg, the ℓ-th (for $\ell \in [0, \mu)$) user's signature on behalf of L with event-id *event* is generated as follows.

1. Compute $\mathbf{g}' = \mathcal{H}_2(event)$, and this implies a temporary instance of $\mathcal{CMT}$ with commitment key $ck' = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}', q)$
2. Compute $\eta = \mathbf{Com}_{ck'}(0, \mathbf{w}_\ell) = \langle \mathbf{g}', \mathbf{w}_\ell \rangle$
3. Parse ℓ as its binary expression $\ell = (\ell_1, \ldots, \ell_\tau)$
4. If the number of iterations from Step 4 to Step 9 reaches r', abort
5. For l from 1 to r
   - For j from 1 to τ
     * sample $\mathbf{r}_{l,j} \leftarrow [-\mathcal{B}, \mathcal{B}]^{m_2 n}$, $a_{l,j} \leftarrow D_s^n$, $\mathbf{s}_{l,j} \leftarrow D_{s'}^{m_2 n}$, $\mathbf{t}_{l,j} \leftarrow D_{s''}^{m_2 n}$
     * compute $c_{l,j} = \mathbf{Com}_{ck}(\ell_{l,j}, \mathbf{r}_{l,j})$
     * compute $d_{l,j} = \mathbf{Com}_{ck}(a_{l,j}, \mathbf{s}_{l,j})$
     * compute $e_{l,j} = \mathbf{Com}_{ck}(a_{l,j}\ell_{l,j}, \mathbf{t}_{l,j})$
   - For k from 0 to $\tau - 1$
     * $\rho_{l,k} \leftarrow D_{s'\sqrt{\tau^{-1}}}^{m_2 n}$
     * compute $v_{l,k} = (\sum_{i=0}^{\mu-1} u_i p_{i,l,k}) + \mathbf{Com}_{ck}(0, \rho_{l,k})$
     * compute $v'_{l,k} = \mathbf{Com}_{ck'}(0, \rho_{l,k}) = \langle \mathbf{g}', \rho_{l,k} \rangle$
   - Let $\mathrm{Cmt}_l = (c_{l,j}, d_{l,j}, e_{l,j}, v_{l,j-1}, v'_{l,j-1})_{j=1}^{\tau}$
6. Compute $\mathbf{x} = (x_1, \ldots, x_r) = \mathcal{H}_1(pp, \mathrm{msg}, L, (\mathrm{Cmt}_l)_{l=1}^r, \eta, event)$
7. For l from 1 to r
   - For j from 1 to τ, compute
     * $f_{l,j} = \ell_{l,j} \cdot x_l + a_{l,j}$
     * $\mathbf{y}_{l,j} = x_l \cdot \mathbf{r}_{l,j} + \mathbf{s}_{l,j}$
     * $\mathbf{z}_{l,j} = (x_l - f_{l,j}) \cdot \mathbf{r}_{l,j} + \mathbf{t}_{l,j}$
   - Compute $\mathbf{v}_l = x_l^\tau \cdot \mathbf{w}_\ell - \sum_{k=0}^{\tau-1} x_l^k \cdot \rho_{l,k}$
8. Let $\mathbf{f}_l = (f_{l,1} \| \ldots \| f_{l,\tau})$, $\mathbf{y}_l = (\mathbf{y}_{l,1} \| \ldots, \| \mathbf{y}_{l,\tau} \| \mathbf{v}_l)$, $\mathbf{z}_l = (\mathbf{z}_{l,1} \| \ldots \| \mathbf{z}_{l,\tau})$, $\mathbf{c}_l = x_l \cdot (\ell_{l,1} \| \ldots \| \ell_{l,\tau})$, $\mathbf{c}'_l = (x_l \cdot \mathbf{r}_{l,1} \| \ldots \| x_l \cdot \mathbf{r}_{l,\tau} \| x_l^\tau \cdot \mathbf{w}_\ell)$, $\mathbf{c}''_l = ((x_l - f_{l,1}) \cdot \mathbf{r}_{l,1} \| \ldots \| (x_l - f_{l,\tau}) \cdot \mathbf{r}_{l,\tau})$, for $l \in [1, r]$.
9. Denoting $\mathbf{f} = (\mathbf{f}_1 \| \ldots \| \mathbf{f}_r)$, $\mathbf{y} = (\mathbf{y}_1 \| \ldots \| \mathbf{y}_r)$, $\mathbf{z} = (\mathbf{z}_1 \| \ldots, \| \mathbf{z}_r)$, $\mathbf{c} = (\mathbf{c}_1 \| \ldots \| \mathbf{c}_r)$, $\mathbf{c}' = (\mathbf{c}'_1 \| \ldots \| \mathbf{c}'_r)$, $\mathbf{c}'' = (\mathbf{c}''_1 \| \ldots \| \mathbf{c}''_r)$, go back to Step 4 with probability

$$1 - \frac{D_s^{\tau nr}(\mathbf{f})}{MD_{\mathbf{c},s}^{\tau n}(\mathbf{f})} \cdot \frac{D_{s'}^{(\tau+1)m_2 nr}(\mathbf{y})}{MD_{\mathbf{c}',s'}^{(\tau+1)m_2 nr}(\mathbf{y})} \cdot \frac{D_{s''}^{\tau m_2 nr}(\mathbf{z})}{MD_{\mathbf{c}'',s''}^{\tau m_2 nr}(\mathbf{z})} \ .$$

10. Let $\mathrm{Rsp}_l = (\mathbf{f}_l, \mathbf{y}_l, \mathbf{z}_l)$
11. Publish the signature $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$, the ring L, the event-id *event*, and the message msg

– **Vfy**$(\mathrm{msg}, event, L, \sigma)$: On input a signature σ, the corresponding ring L, the event-id *event* and the message msg, this algorithm does as follows to test the validity of σ.

- Check if $\mathbf{x} = (x_1, \ldots, x_r) = \mathcal{H}_1(pp, \mathrm{msg}, L, (\mathrm{Cmt}_l)_{l=1}^r, \eta, event)$. Return 0 if it is not

- Compute $\mathbf{g}' = \mathcal{H}_2(event)$ to obtain the temporary commitment key ck'
- For l from 1 to r
  - For j from 1 to τ, consider the following conditions
    - $\|f_{l,j}\| \leq s\sqrt{n}$
    - $\|\mathbf{y}_{l,j}\| \leq s'\sqrt{m_2 n}$
    - $\|\mathbf{z}_{l,j}\| \leq s''\sqrt{m_2 n}$
    - $x_l \cdot c_{l,j} + d_{l,j} = \mathbf{Com}_{ck}(f_{l,j}, \mathbf{y}_{l,j})$
    - $(x_l - f_{l,j}) \cdot c_{l,j} + e_{l,j} = \mathbf{Com}_{ck}(0, \mathbf{z}_{l,j})$

    If any of them does not hold, output 0 and abort.
  - If $\|\mathbf{v}_l\| \leq s'\sqrt{m_2 n}$ is not satisfied, output 0 and abort.
  - If the equation $\eta x_l^\tau + \sum_{k=0}^{\tau-1} v'_{l,k}(-x_l^k) = \mathbf{Com}_{ck'}(0, \mathbf{v}_l) = \langle \mathbf{g}', \mathbf{v}_l \rangle$ does not hold, output 0 and abort.
  - If $\sum_{i=0}^{\mu-1}(u_i \prod_{j=1}^{\tau} f_{l,j,i_j}) + \sum_{k=0}^{\tau-1} v_{l,k}(-x_l^k) = \mathbf{Com}_{ck}(0, \mathbf{v}_l)$ does not hold, output 0 and abort.

  If all above conditions satisfy, return 1 to accept the signature.

– **Link**$(event, \mathrm{msg}_1, L_1, \sigma_1, \mathrm{msg}_2, L_2, \sigma_2)$: For two accepting signatures $\sigma_1 = (\ldots, \eta_1)$ and $\sigma_2 = (\ldots, \eta_2)$ on the same event-id $event$, if $\eta_1 = \eta_2$, return 1 for concluding that the signatures are linked; otherwise, return 0.

5.3 Security Proofs

Theorem 6. *$\mathcal{LRS}$ with parameters in Table 4 is statistical correct with a negligible correctness error. The expected number of iterations for* **Sign** *is $M^3 = O(1)$.*

Proof. If $\mathcal{LRS}$.**Sign** outputs a signature σ, then σ could pass the algorithm $\mathcal{LRS}$.**Vfy**, because of the discussion on the completeness of Σ_2 in Sect. 4.3. With the specific parameter settings, and the strategy of rejection samplings in $\mathcal{LRS}$.**Sign**, the aborting probability of running the underlying protocol Σ_2 r times in parallel is $1 - \left(\frac{1-\epsilon}{M}\right)^3$. Thus, for a maximum number of iterations $r' = \frac{-n}{\log(1-(1-\epsilon)^3/M^3)}$, the probability that all iterations fail to output a signature is

$$(1 - (1-\epsilon)^3/M^3)^{\frac{-n}{\log(1-(1-\epsilon)^3/M^3)}} = 2^{\log(1-(1-\epsilon)^3/M^3) \cdot \frac{-n}{\log(1-(1-\epsilon)^3/M^3)}} = 2^{-n} \ .$$

Consequently, the number of iterations in $\mathcal{LRS}$.**Sign** reaches r' with negligible probability. Since each iteration outputs an accepting signature with probability $(1-\epsilon)^3/M^3$, the expected number of iterations for $\mathcal{LRS}$.**Sign** is $M^3/(1-\epsilon)^3 \approx 22 = O(1)$. □

Theorem 7. *$\mathcal{LRS}$ with parameters in Table 4 is of statistically weak anonymity if $\mathcal{CMT}$ is statistically hiding and computationally binding.*

Proof. Notice that Σ_2 is SHVZK. Subsequently, if $\mathcal{LRS}$.**Sign** outputs a signature $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$, then conditioned on the challenge $\mathbf{x} \in \mathcal{C}^r$, initial messages $(\mathrm{Cmt}_l)_{l=1}^r$ and responses $(\mathrm{Rsp}_l)_{l=1}^r$ are statistically independent of the secret information of the real signer. This fact was also stated in Theorem

1 of [14]. Consequently, if an adversary is able to break the statistically weak anonymity, its advantage mainly comes from the linking tag η.

Denote $\mathbf{g} = (g_1, \ldots, g_{m_2})$, $\mathbf{G} = (Rot(g_1), \ldots, Rot(g_{m_2}))$, $\mathbf{g}' = (g'_1, \ldots, g'_{m_2})$, $\mathbf{G}' = (Rot(g'_1), \ldots, Rot(g'_{m_2}))$. By the parameter settings and Theorem 1, we can see that for all but at most $\mathrm{negl}(n)$ fraction of $\mathbf{g} \in R_q^{m_2}$, the columns of $\mathbf{G} \in \mathbb{Z}_q^{n \times m_2 n}$ generates $\mathbb{Z}_q^n$, so does $\mathbf{G}'$. Since $m_2 > 2$, then with an overwhelming probability, there exists a $\mathbf{w} \in [-\mathcal{B}, \mathcal{B}]^{m_2 n}$ that satisfies $\binom{\mathbf{G}}{\mathbf{G}'}\mathbf{w}^T = \binom{\mathrm{vec}^T(u)}{\mathrm{vec}^T(\eta)}$, where $u \in R_q$ is an arbitrary verifying key. Consequently, if the adversary cannot corrupt the members of the signature, η is useless for it to determine the real signer with overwhelming probability, resulting the theorem. □

The ideas to prove the remaining theorems were originated in [14] and was extended to handle the specific problems caused by lattices in [10]. Before we begin, we introduce some notations employed in [10].

Let Ψ be the set of all random tapes that could be used by a PPT adversary $\mathcal{A}$ and Φ be the set of all random tapes defining the random oracle $\mathcal{RO}$. Let $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,r})$ be the output of j-th random oracle query. We partition Φ into Φ_{j-}, $\mathbf{x}_j$ and Φ_{j+} so that Φ_{j-}, Φ_{j+} represent the sets of random tapes defining the random oracle outputs up to j-th query (*i.e.*, $\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}$) and after j-th query (*i.e.*, $\mathbf{x}_{j+1}, \ldots, \mathbf{x}_Q$), respectively. Therefore, the tuple $(\phi_{j-}, \mathbf{x}_j, \phi_{j+})$ defines all the random oracle outputs. Notice that $\mathcal{A}$ fixes its random tape with $\psi \leftarrow \Psi$ when it is initiated, and fills in additional random values in its random tape by interacting with $\mathcal{RO}$. If $Algo$ is a probabilistic algorithm, by writing $Algo[\mathbf{x}]$, we omit the input of the algorithm (which is clear from the context) and emphasize the value of its random tape is $\mathbf{x}$.

We also notice that by the assumption on statistically hiding and computationally binding, Σ_2 in Sect. 4.3 is SHVZK and $(\tau+1)$-special sound, according to Theorem 5.

Theorem 8. *$\mathcal{LRS}$ with parameters in Table 4 is unforgeable in the random oracle model, if $\mathcal{CMT}$ with respect to $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ is statistically hiding and computationally binding, where $\mathbf{g}, \mathbf{h}$ are chosen independently and uniformly.*

Proof. The key point of the proof is to simulate a unforgeable game (using SHVZK) with a PPT adversary $\mathcal{A}$, which is able to break the unforgeability of $\mathcal{LRS}$. In the simulated game, we try to rewind $\mathcal{A}$ to obtain $\tau + 1$ successful forgeries on distinct challenges. Then the $(\tau + 1)$-soundness extractor for Σ_2 gives us a chance to open a commitment (verifying key) in a different way.

Consider a PPT adversary $\mathcal{A}$ which runs within $\mathrm{poly}(n)$ steps and makes at most $q_V(n)$, $q_S(n)$, $q_H(n)$ honest queries to $\mathcal{JO}$, $\mathcal{SO}$ and the random oracle $\mathcal{RO}$ defined in Sect. 2.4, respectively, is able to output a valid forgery with probability $\varepsilon = 1/\mathrm{poly}(n)$. Here, a valid forgery means that the resulting signature without any corrupted verifying keys is valid and it is not generated by $\mathcal{SO}$. Let $Q = q_S + q_H$ be the maximum number of random oracle queries made by $\mathcal{A}$. Depending on this, we construct a PPT algorithm $\mathcal{A}'$ to break the binding property of $\mathcal{CMT}$.

$\mathcal{A}'$ initiates the attack by running $\mathcal{LRS}.\mathbf{Setup}(n, \mu)$ to generate the global parameters of $\mathcal{LRS}$ and sends these parameters to $\mathcal{A}$. $\mathcal{A}'$ models $\mathcal{H}_1$ as a random oracle and does as follows.

1. pick at random $t \leftarrow [1, q_V]$.
2. pick $\mathbf{w}_t \leftarrow [-\mathcal{B}, \mathcal{B}]^{m_2 n}$ and compute $vk_t = \mathbf{Com}(1, \mathbf{w}_t)$.
3. Pick $j \leftarrow [1, Q]$.
4. Pick $\psi \leftarrow \Psi$.
5. Pick $(\phi_{j-}, \mathbf{x}_j^0, \phi_{j+}) \leftarrow \Phi_{j-} \times \mathcal{C}^r \times \Phi_{j+}$.
6. Run $\mathcal{A}[\psi, \phi_{j-}, \mathbf{x}_j^0, \phi_{j+}]$ with access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle $\mathcal{RO}[\phi_{j-}, \mathbf{x}_j^0, \phi_{j+}]$ that are simulated by $\mathcal{A}'$.
   - $\mathcal{JO}(\perp)$. Whenever $\mathcal{A}$ queries $\mathcal{JO}$, $\mathcal{A}'$ runs $\mathcal{LRS}.\mathbf{KGen}(pp)$ with fresh random coins and sends back the verifying key, except for the t-th query in which it returns vk_t.
   - $\mathcal{CO}(vk_j)$. Only if $\mathcal{A}$ queries $\mathcal{CO}(vk_t)$, $\mathcal{A}'$ aborts (Type I). Otherwise, it returns the corresponding signing key.
   - $\mathcal{SO}(vk_j, \mathrm{msg}, event, L)$.
     - If $vk_j \neq vk_t$, $\mathcal{A}'$ runs $\mathcal{LRS}.\mathbf{Sign}(sk_i, \mathrm{msg}, event, L)$ to obtain a signature σ, but the challenge vector $\mathbf{x}$ is generated by querying $\mathcal{RO}$. Return σ.
     - If $vk_j = vk_t$, $\mathcal{A}'$ makes a special queries to $\mathcal{RO}$. It directly picks the current fresh output of $\mathcal{RO}$. Let the random challenge vector be $\mathbf{x}$. $\mathcal{A}'$ uses the SHVZK simulator of Σ_2 in Sect. 4.3 to simulate a proof $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$ (only the simulation of non-aborted protocols is used here). However, if $(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event)$ has been queried to $\mathcal{RO}$ before, $\mathcal{A}'$ aborts (Type II). Otherwise, return σ, and program $\mathcal{RO}$ to have $\mathcal{H}_1(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event) = \mathbf{x}$.
   - $\mathcal{RO}(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event)$. If the tuple has been queried before, return the corresponding $\mathbf{x}$ programmed in the random oracle. Otherwise, $\mathcal{A}'$ returns the current fresh output of $\mathcal{RO}$, denoted by $\mathbf{x}$, and programs $\mathcal{RO}$ to have $\mathcal{H}_1(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event) = \mathbf{x}$.

   If $\mathcal{A}$ outputs a forgery σ^0 using j-th random oracle query output $\mathbf{x}_j^0$, fix ψ and ϕ_{j-}, so that the strategy for $\mathcal{A}$ to forge before it obtains the j-th random oracle query is fixed. Otherwise, $\mathcal{A}'$ abort.
7. For $i \in [1, \mathcal{N}]$, pick $\phi_i' \leftarrow \Phi_{j+}$ and $\mathbf{x}_j^i \leftarrow \mathcal{C}^r$ independently. ϕ_{j-}, $\mathbf{x}_j^i$, ϕ_i' constitute a new random tape of $\mathcal{RO}$.
8. For $i \in [1, \mathcal{N}]$,
   - run $\mathcal{A}[\psi, \phi_{j-}, \mathbf{x}_j^i, \phi_i']$ with access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle $\mathcal{RO}[\phi_{j-}, \mathbf{x}_j^i, \phi_i']$.
   - $\mathcal{A}$ outputs a forgery σ^i, which may not using $\mathbf{x}_j^i$.
9. Denoting $\mathbf{x}_j^i$ by $(x_{j,1}^i, \ldots, x_{j,r}^i)$, check if there exists $k \in [1, r]$ and $S^* \subseteq [0, \mathcal{N}]$ with $|S^*| = \tau + 1$ such that $G^* \stackrel{\text{def}}{=} \{x_{j,k}^u : u \in S^*\}$ contains $\tau + 1$ distinct challenges and σ^u is a valid forgery using $\mathbf{x}_j^u$, for all $u \in S^*$. Abort if it is not.

10. Run $(\tau+1)$-special soundness extractor on input $\{\sigma^u\}_{u\in S^*}$ to extract an opening of $2^\tau \cdot vk_{t'}$ denoted by $(0, \mathbf{w}'_{t'})$ for some $1 \le t' \le q_V(\lambda)$, where $\|\mathbf{w}'_{t'}\|_\infty \le (\tau+1)\cdot n^{\tau-0.5}\cdot 2s'\sqrt{m_2 n}$.
11. If $t \neq t'$, aborts. Otherwise, $(2^\tau \cdot 1, 2^\tau \cdot \mathbf{w}_t)$ and $(0, \mathbf{w}'_{t'})$ are a pair of distinct openings for the commitment $2^\tau vk_t$.

If $\mathcal{A}'$ succeed in reach step 11 and $t = t'$, there is no Type I abort as the forged signatures do not use a ring with corrupted verifying keys. Moreover, the view of $\mathcal{A}$ in the simulated game is the same as in a real game except for:

- vk_t is a commitment to 1 in the simulation by $\mathcal{A}'$ whereas it is a commitment to 0 in the real game. By the $\mathrm{negl}(n)$-statistical hiding of the commitment scheme, this reduces the success probability of $\mathcal{A}$ by at most $\mathrm{negl}(n)$.
- Since $\mathcal{SO}$ was queried at most q_S times, the maximum number of running the $SHVZK$ simulator to generate a signature is q_S. Hence, the statistical distance between the distribution of signing oracle simulator and that of the real signing oracle is at most $q_S \cdot \mathrm{negl}(n)$.
- Since $\mathcal{CMT}$ is statistically hiding and computationally binding, when $\mathcal{A}'$ needs to run the SHVZK simulator to generate a signature, its abort (type II) with probability at most $\mathrm{negl}(n)$. As $\mathcal{A}$ makes at most Q random oracle queries, $\mathcal{A}'$ aborts with probability at most $Q \cdot \mathrm{negl}(n)$.

By the simulation statistical distance argument above, each run of $\mathcal{A}$ with vk_t and signing oracle simulated by $\mathcal{A}'$ succeeds with probability $\tilde{\varepsilon} \ge \varepsilon - Q\cdot\mathrm{negl}(n)$.

For $j \in [1, Q]$, let E_j be the event that $\mathcal{A}[\psi, \phi_{j_-}, \mathbf{x}_j, \phi_{j_+}]$ outputs a valid forgery signature using the challenge $\mathbf{x}_j$, where $\psi, \phi_{j_-}, \mathbf{x}_j, \phi_{j_+}$ are chosen uniformly and independently. Since $\mathcal{A}$ has probability $\tilde{\varepsilon}$ to output a valid forgery, then by an averaging argument on j, there exists a $j^* \in [1, Q]$,

$$\Pr_{(\psi,\phi_{j_-},\mathbf{x}_j,\phi_{j_+})\leftarrow \Psi\times\Phi_{j_-}\times\mathcal{C}^r\times\Phi_{j_+}}[E_j \mid j = j^*] \ge \tilde{\varepsilon}/Q \ .$$

For such a specific j^*, define $\Psi \times \Phi_{j^*_-} = X$, $\mathcal{C}^r \times \Phi_{j^*_+} = Y$ and let A be the subset of $X \times Y$ that yields E_{j^*} occurs, *i.e.*, $A = \{(a,b) \in X \times Y : E_{j^*} \text{ occurs}\}$. Then by the probability of E_{j^*} occurring, we have $\Pr_{(a,b)\leftarrow X\times Y}[(a,b) \in A] \ge \tilde{\varepsilon}/Q$. Subsequently, let $B = \{a \in X : \Pr_{b\leftarrow Y}[(a,b) \in A] \ge \tilde{\varepsilon}/(2Q)\}$. Then according to Lemma 9,

$$\Pr_{a\leftarrow X}[x \in B] \ge \tilde{\varepsilon}/(2Q) \ ,$$

and for all $a \in B$,

$$\varepsilon' \stackrel{\text{def}}{=} \Pr_{b\leftarrow Y}[(a,b) \in A] \ge \tilde{\varepsilon}/(2Q) \ .$$

For notation simplicity in the further discussion, let $Y = \mathcal{C}^r \times Z$ and denote a challenge $\mathbf{x}$ by $(x_1, \ldots, x_r)$. Conditioned on $a \in B$, define the conditional probability regarding to a $c \in \mathcal{C}$ by

$$p_i(c) = \Pr_{(\mathbf{x},b)\leftarrow \mathcal{C}^r\times Z}[(a,\mathbf{x},b) \in A \wedge x_i = c \mid a \in B] \ ,$$

which is the conditional probability that E_{j^*} occurs and c equals to the i-th entry of the challenge vector in the forged signature, when $a \in B$. Since $\varepsilon' \geq \tilde{\varepsilon}/(2Q) > (\tau/|\mathcal{C}|^r) = \mathrm{negl}(n)$ and $|\mathcal{C}| > \tau$, from Lemma 10, there exists an $i^* \in [1, r]$ and $G \subseteq \mathcal{C}$ with $|G| = \tau + 1$, such that for all $c \in G$, $p_{i^*}(c) \geq \frac{\varepsilon' - (\tau/|\mathcal{C}|^r)}{|\mathcal{C}| - \tau} \cdot r \stackrel{\text{def}}{=} p$.

Let $\mathcal{N} \stackrel{\text{def}}{=} (\tau+1) \cdot p^{-1} - 1$. For such an i^* and a set G, conditioned on $a \in B$, if we independently picks elements $(\mathbf{x}^{(0)}, b_0), \ldots, (\mathbf{x}^{(\mathcal{N})}, b_{\mathcal{N}}) \leftarrow \mathcal{C}^r \times Z$, and runs $\mathcal{A}[a, \mathbf{x}^{(k)}, b_k]$, for $k \in [0, \mathcal{N}]$, then the probability that "$\exists c \in G$, c is not the i^*-th entry of $\mathbf{x}_{j^*}$ used in a valid forged signature" is

$$\begin{aligned}
&\Pr[\exists c \in G, \forall (\mathbf{x}^{(k)}, b_k), (a, \mathbf{x}, b_k) \notin A \vee x_{i^*} \neq c \mid a \in B] \\
&\leq |G| \cdot \max(\Pr[c \in G, \forall (\mathbf{x}^{(k)}, b_k), (a, \mathbf{x}, b_k) \notin A \vee x_{i^*} \neq c \mid a \in B]) \\
&= (\tau+1)(1-p)^{\mathcal{N}+1} \\
&= (\tau+1)((1-p)^{-p^{-1}})^{-(\tau+1)} \\
&= (\tau+1)e^{-(\tau+1)} \ .
\end{aligned}$$

Consequently, the probability that "$\forall c \in G$, c is the i^*-th entry of x_{j^*} used in a valid forged signature" is

$$\begin{aligned}
\zeta &\stackrel{\text{def}}{=} \Pr[\forall c \in G, \exists (\mathbf{x}^{(k)}, b_k), (a, \mathbf{x}, b_k) \in A \wedge x_{i^*} = c \mid a \in B] \\
&\geq 1 - \Pr[\exists c \in G, \forall (\mathbf{x}^{(k)}, b_k), (a, \mathbf{x}, b_k) \notin A \vee x_{i^*} \neq c \mid a \in B] \\
&= 1 - (\tau+1)e^{-(\tau+1)} \geq 7/10 \ ,
\end{aligned}$$

where the last inequality is because that $(\tau+1)e^{-(\tau+1)}$ is monotone decreasing, $\tau \geq 1$, and $e < 3$.

Since $|\mathcal{C}| = 2n$, $r = \frac{n}{\log(2n) - \log\tau}$, $\tau = \log\mu$ are all upper-bounded by $\mathrm{poly}(n)$, and

$$\begin{aligned}
(\varepsilon' - (\tau/|\mathcal{C}|)^r)^{-1} &\leq \left(\frac{\tilde{\varepsilon}}{2Q} - \mathrm{negl}(n)\right)^{-1} \\
&\leq \left(\frac{\varepsilon - Q \cdot \mathrm{negl}(n)}{2Q} - \mathrm{negl}(n)\right)^{-1} \\
&= \mathrm{poly}(n) \ ,
\end{aligned}$$

we have $\mathcal{N} + 1 = (\tau+1) \cdot p^{-1} = (\tau+1)\frac{(|\mathcal{C}|-\tau)\cdot r}{\varepsilon' - (\tau/|\mathcal{C}|)^r} \leq \mathrm{poly}(n)$.

Now, by $(\tau+1)$-special soundness of Σ_2, we can use the set G to extract an opening of $2^\tau vk_{t'}$ to $(0, \mathbf{w}'_{t'})$ for some $t' \in [1, q_V]$. By the $\mathrm{negl}(n)$-statistical hiding property of the commitment scheme, $t' = t$ with probability at least $\frac{1}{q_V} - \mathrm{negl}(n)$. On the other side, $j = j^*$ with probability $1/Q$. Hence, $\mathcal{A}'$ succeeds to output a pair of distinct openings with probability

$$\Pr[j = j^*] \cdot \Pr[a \in B] \cdot \zeta \cdot \Pr[t = t'] \geq \frac{1}{Q} \cdot \frac{\tilde{\varepsilon}}{2Q} \cdot \frac{7}{10} \cdot \left(\frac{1}{q_V} - \mathrm{negl}(n)\right) = \frac{1}{\mathrm{poly}(n)} \ .$$

As a result, $\mathcal{A}'$ running within $\mathrm{poly}(n)$ steps, breaks the binding property of $\mathcal{CMT}$ with non-negligible probability. Consequently, $\mathcal{LRS}$ is unforgeable if $\mathcal{CMT}$ is computational binding. □

Theorem 9. *$\mathcal{LRS}$ with parameters in Table 4 is of linkability in the random oracle model, if $\mathcal{CMT}$ with respect to $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ is statistically hiding and computationally binding, where $\mathbf{g}, \mathbf{h}$ are chosen independently and uniformly. More specifically, if a PPT adversary grasps only one key pair (sk, vk), then it has negligible probability to generate an accepting signature $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$ with respect to the ring L, event-id event, such that $\eta \neq \langle \mathcal{H}_2(event), sk \rangle$.*

Proof. We first observe such a fact. Because of the unforgeability of $\mathcal{LRS}$, any PPT adversary $\mathcal{A}$ has negligible probability to generate an accepting signature, if it employs a ring L such that the corrupted verifying key is not in L. Consequently, for the sake of contradiction, we assume the PPT adversary $\mathcal{A}$

- Makes q_V, q_S, q_H queries to $\mathcal{JO}$, $\mathcal{SO}$, $\mathcal{RO}$, respectively.
- Queries $\mathcal{CO}$ one time. Denote the corrupted key pair by $(\overline{vk}, \overline{sk})$.
- Generates a valid signature $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$ on behalf of a ring L and event-id *event*, with probability $\varepsilon = \frac{1}{\mathrm{poly}(n)}$. Here a valid signature means σ was not generated by $\mathcal{SO}$, $\overline{vk} \in L$, $\eta \neq \langle \mathcal{H}_2(event), \overline{sk} \rangle$.

With this, we construct a PPT algorithm $\mathcal{A}'$ using the idea in the proof of unforgeability to violate the assumptions in the current theorem.

1. Pick $j \leftarrow [1, Q]$.
2. Pick $\psi \leftarrow \Psi$.
3. Pick $(\phi_{j-}, \mathbf{x}_j^0, \phi_{j+}) \leftarrow \Phi_{j-} \times \mathcal{C}^r \times \Phi_{j+}$.
4. Run $\mathcal{A}[\psi, \phi_{j-}, \mathbf{x}_j^0, \phi_{j+}]$ with access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle $\mathcal{RO}[\phi_{j-}, \mathbf{x}_j^0, \phi_{j+}]$ that are simulated by $\mathcal{A}'$.
   - $\mathcal{JO}(\perp)$. When $\mathcal{A}$ queries $\mathcal{JO}$, $\mathcal{A}'$ runs $\mathcal{LRS}.\mathbf{KGen}(pp)$ with fresh random coins and sends back the verifying key.
   - $\mathcal{CO}(vk_j)$. If vk_j is generated by $\mathcal{JO}$, return the corresponding signing key. $\mathcal{CO}$ is only allowed to be queried one time.
   - $\mathcal{SO}(vk_j, \mathrm{msg}, event, L)$. $\mathcal{A}'$ runs $\mathcal{LRS}.\mathbf{Sign}(sk_i, \mathrm{msg}, event, L)$ to obtain a signature σ, but the challenge vector $\mathbf{x}$ is generated by querying $\mathcal{RO}$. Return σ.
   - $\mathcal{RO}(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event)$. If the tuple has been queried before, return the corresponding $\mathbf{x}$ programmed in the random oracle. Otherwise, $\mathcal{A}'$ returns the current fresh output of $\mathcal{RO}$, and programs $\mathcal{RO}$ to have $\mathcal{H}_1(pp, \mathrm{msg}, L, (\mathrm{Cmt}_i)_{i=1}^r, \eta, event) = \mathbf{x}$.

   If $\mathcal{A}$ outputs a signature $\sigma^0 = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}_j^0, (\mathrm{Rsp}_l^0)_{l=1}^r, \eta)$, and the linking tag $\eta \neq \langle \mathcal{H}_2(event), \overline{sk} \rangle$, fix ψ, and ϕ_{j-}. Otherwise, abort.
5. For $i \in [1, \mathcal{N}]$, pick $\phi_i' \leftarrow \Phi_{j+}$ and $\mathbf{x}_j^i \leftarrow \mathcal{C}^r$ independently.
6. For $i \in [1, \mathcal{N}]$,
   - run $\mathcal{A}[\psi, \phi_{j-}, \mathbf{x}_j^i, \phi_i']$ with access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle $\mathcal{RO}[\phi_{j-}, \mathbf{x}_j^i, \phi_i']$.
   - $\mathcal{A}$ outputs a signature $\sigma^i = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{y}^i, (\mathrm{Rsp}_l^i)_{l=1}^r, \eta)$, where $\mathbf{y}^i$ may not equal to $\mathbf{x}_j^i$.

7. Denoting $\mathbf{x}_j^i$ by $(x_{j,1}^i, \ldots, x_{j,r}^i)$, check if there exists $k \in [1, r]$ and $S^* \subseteq [0, \mathcal{N}]$ with $|S^*| = \tau + 1$ such that $G^* \stackrel{\text{def}}{=} \{x_{j,k}^u : u \in S^*\}$ contains $\tau + 1$ distinct challenges and σ^u is a valid signature using $\mathbf{x}_j^u$ with linking tag $\eta \neq \langle \mathcal{H}_2(event), \overline{sk} \rangle$, for all $u \in S^*$. Abort if it is not.
8. Run $(\tau + 1)$-special soundness extractor on input $\{\sigma^u\}_{u \in S^*}$ to extract a witness $(\ell', \mathbf{w}')$, where $\|\mathbf{w}'\|_\infty \leq (\tau+1) \cdot n^{\tau - 0.5} \cdot 2s' \sqrt{m_2 n}$, and $\ell' \in [0, 2^\tau)$. By the discussion on the knowledge extractor, we have $2^\tau vk_{\ell'} = \mathbf{Com}(0, \mathbf{w}') = \mathbf{Com}(0, 2^\tau sk_{\ell'})$, and $2^\tau \eta = \langle \mathcal{H}_2(event), \mathbf{w}' \rangle$.
Case 1. If $\mathbf{w}' \neq 2^\tau sk_{\ell'}$, this yields a contraction to that $\mathcal{CMT}$ is computationally binding, as $sk_{\ell'}$ was generated by $\mathcal{A}'$.
Case 2. If $\mathbf{w}' = 2^\tau sk_{\ell'}$ and $vk_{\ell'} \neq \overline{vk}$, then as 2 is invertible in R_q, we have $2^{-\tau}\mathbf{w}' = sk_{\ell'}$, which means $\mathcal{A}$ has the knowledge of $sk_{\ell'}$ and yields a contradiction to that $\mathcal{A}$ only corrupted one key pair.
Case 3. If $\mathbf{w}' = 2^\tau sk_{\ell'}$ and $vk_{\ell'} = \overline{vk}$, we have $2^\tau \overline{sk} = \mathbf{w}'$. However, as $\eta \neq \langle \mathcal{H}_2(event), \overline{sk} \rangle$, we can conclude that

$$\langle \mathcal{H}_2(event), \mathbf{w}' \rangle = 2^\tau \eta \neq \langle \mathcal{H}_2(event), 2^\tau \overline{sk} \rangle \ ,$$

so that $\mathbf{w}' \neq 2^\tau \overline{sk}$, which yields a contradiction to the condition of this case. Thus, conditioned on $\mathcal{A}$ only querying on $\mathcal{CO}$ one time, and $\{\sigma^u\}_{u \in S^*}$ are valid signature using $\mathbf{x}_j^u$ with linking tag $\eta \neq \langle \mathcal{H}_2(event), \overline{sk} \rangle$, **Case 2** and **Case 3** are impossible. We obtain from **Case 1** a contraction to the computational binding property of $\mathcal{CMT}$.

The essential ideas of the PPT algorithm $\mathcal{A}'$ are the same as the algorithm in the proof of unforgeability, so that the discussion on the probability of $\mathcal{A}'$ obtaining $\tau + 1$ valid signatures to extract a knowledge is similar to that of the proof of unforgeability. The differences are as follows

- The game simulated by $\mathcal{A}'$ is the same as a real game in the view of $\mathcal{A}$.
- $\mathcal{A}'$ does not need to embed a fake verifying key (*i.e.*, $vk_t = \mathbf{Com}(1, \mathbf{w}_t)$) in the game, so that to extract the knowledge of vk_t is not necessary.

Consequently, we directly give the probability for $\mathcal{A}'$ to break the binding property

$$\Pr[j = j^*] \cdot \Pr[(\psi, \phi_{j_-}) \in S] \cdot \zeta \geq \frac{1}{Q} \cdot \frac{\varepsilon}{2Q} \cdot \frac{7}{10} = \frac{1}{\text{poly}(n)}$$

□

Theorem 10. *$\mathcal{LRS}$ with parameters in Table 4 is nonslanderable in the random oracle mode, if $\mathcal{CMT}$ with respect to $ck = (n, m_1, m_2, \beta, \mathbf{h}, \mathbf{g}, q)$ is computationally binding and statistically hiding, where $\mathbf{g}, \mathbf{h}$ are chosen independently and uniformly.*

Proof. This is implied by the unforgeability and linkability. Depending on Theorem 8, any PPT adversary has negligible probability to generate a signature on behalf of L such that it does not know one of the signing keys of the verifying keys in L. Consequently, conditioned on a PPT adversary $\mathcal{A}$ outputs an

accepting signature $\sigma = ((\mathrm{Cmt}_l)_{l=1}^r, \mathbf{x}, (\mathrm{Rsp}_l)_{l=1}^r, \eta)$ on behalf of L and event-id $event$, it holds some (sk, vk), such that $vk \in L$ with overwhelming probability. Then by the proof of Theorem 9, it is with overwhelming probability that $\eta = \langle \mathcal{H}_2(event), sk \rangle$. Combining the above discussions, if for some $vk' \in L$ that is not corrupted by $\mathcal{A}$ and $sk' \neq sk$ is the corresponding signing key, $\mathcal{A}$ has negligible probability to generate an accepting signature such that $\eta = \langle \mathcal{H}_2(event), sk' \rangle$, resulting the theorem. □

6 APQC Based on Linkable Ring Signatures

In CryptoNote, the author suggested using stealth addresses to protect the privacy of receivers in all transactions. A stealth address is a one-time address (a verifying key which is also called a destination key) for a receiver to receive coins. It is generated by the sender of a transaction, and only the real receiver could determine the one-time address and recover the corresponding signing key.

In this section, we will introduce a key-generation protocol to handle stealth addresses. By combining this protocol and the linkable ring signature presented in the previous section, we describe a standard transaction of APQC.

6.1 The Public-key Encryption from Ideal Lattices

The public encryption scheme we employed in our APQC was proposed by Stehlé *et al.* [38]. The ideal-lattice-based encryption scheme is formalized as a collection of efficient procedures $\mathcal{ES}$=(**Setup**, **KGen**, **Enc**, **Dec**). Due to lack of space, we refer readers to [38] for the details.

The notion of key privacy is formally defined by Bellare *et al.* [5]. It requires that the receiver of a ciphertext is anonymous from the point of view of the adversary. Fortunately, we can deduce from the observation 1 of [15] that the aforementioned encryption scheme $\mathcal{ES}$ is of key privacy.

6.2 Key-Generation Protocol

The key-generation protocol is responsible for three purposes. Firstly, it generates public and private keys for a user that initially joins the cryptocash system. Secondly, if Alice wants to pay coins to Bob, this protocol generates a fresh one-time address for Bob by using the random values chosen by Alice and the public key of Bob. Note that the one-time address is essentially a verifying key of the linkable ring signature scheme. Thirdly, since Alice broadcasts the transaction labeled with the destination address, the key-generation protocol helps Bob to efficiently recognize this transaction and to recover the corresponding signing key.

This protocol is formalized as four efficient procedures $\mathcal{KG}$=(**Setup**, **UKeyGen**, **DKeyGen**, **DKeyRec**) which are short forms for setup, user keys generation, destination keys generation, and destination keys recovery, respectively.

Setup(1^n): On input security parameter, this procedure generates global parameters pp for the whole cryptocash system which means this procedure also runs $\mathcal{LRS}.\mathbf{Setup}(1^n)$ and $\mathcal{ES}.\mathbf{Setup}(1^n)$ as subroutines so that the signature scheme and encryption scheme are accurately initiated (see Sect. 6.1 and Sect. 5.2 for details). Let pp be the public global parameters of the linkable ring signature. Besides that, it chooses a cryptographic hash function $hash : \{0,1\}^* \to \{0,1\}^{2n}$, and a unique event-id $event$ employed in every linkable ring signature of the system, for example, it could be the public parameters of $\mathcal{LRS}$.

UKeyGen(pp): When a user wants to join the cryptocash system, he/she executes this procedure. This procedure first generates the keys for public key encryption scheme $(epk, esk) \leftarrow \mathcal{ES}.\mathbf{KGen}(pp)$. It then generates a partial key pair for the $\mathcal{LRS}$ by randomly picking $\mathbf{w} \in [-\mathcal{B}/2, \mathcal{B}/2]^{m_2 n}$, parsing $\mathbf{w}$ as a vector in $R_q^{m_2}$ and computing $u = \mathbf{Com}(0, \mathbf{w})$. Note that the norm of the partial signing key $\mathbf{w}$ is a little smaller than the original one of the linkable ring signature. (epk, u) and $(esk, \mathbf{w})$ are the public and private keys held by the user.

DKeyGen(pp, epk, u): If Alice wants to send coins to Bob who holds keys (epk, u), $(esk, \mathbf{w})$, she runs the procedure with epk and u. This procedure randomly picks $\mathbf{w}_p \in [-\mathcal{B}/2, \mathcal{B}/2]^{m_2 n}$ and generates the destination key $u_d = \mathbf{Com}(0, \mathbf{w}_p) + u$ for Bob. $\mathbf{w}_p$ is a part of the signing key with respect to the destination key u_d, but no one except Bob can recover the integral signing key (since generalized knapsack functions are one-wayness). This procedure proceeds to pick an AES secret key k uniformly at random. It then computes $c_1 = \mathcal{ES}.\mathbf{Enc}_{epk}(k)$ with the public key encryption and computes $c_2 = \mathbf{AES}_k(hash(epk) \| \mathbf{w}_p)$ with the AES algorithm. Finally, it outputs the destination key u_d, and the auxiliary information c_1, c_2. The process of **DkeyGen** procedure is depicted in Fig. 1.

DKeyRec$(pp, epk, esk, u, \mathbf{w}, (u_d, c_1, c_2))$: Bob runs this procedure to check (u_d, c_1, c_2) of a passing transaction. If it is a transaction with Bob as recipient, it will be that 1) $k = \mathcal{ES}.\mathbf{Dec}_{esk}(c_1)$; 2) $(hash(epk) \| \mathbf{w}_p) = \mathbf{AES}_k(c_2)$. If this procedure finds out that the first part of the plaintext of c_2 is not the hash value of Bob's public encryption key epk, then this procedure aborts and outputs 0. Otherwise, Bob computes $\mathbf{w}_d = \mathbf{w}_p + \mathbf{w}$ and $u'_d = \langle \mathbf{g}, \mathbf{w}_d \rangle$. If $u'_d = u_d$, this procedure outputs 1 and admits the validity of the destination key u_d and its signing key $\mathbf{w}_d$. Since $\|\mathbf{w}_d\|_\infty \leq \|\mathbf{w}_p\|_\infty + \|\mathbf{w}\|_\infty \leq \mathcal{B}$, $\mathbf{w}_d$ is a valid signing key with correspondence to the destination key u_d. The process of this procedure is briefly shown in Fig. 2.

6.3 Transactions

We proceed to introduce transactions in APQC. Let Bob and Alice be two users of our APQC. Bob will runs $\mathcal{KG}.\mathbf{UKeyGen}$ to generates his public and private keys $(epk_{\text{Bob}}, u_{\text{Bob}})$, $(esk_{\text{Bob}}, \mathbf{w}_{\text{Bob}})$, when he initially joins the system. Similarly, $(epk_{\text{Alice}}, u_{\text{Alice}})$, $(esk_{\text{Alice}}, \mathbf{w}_{\text{Alice}})$ are the keys held by Alice. Besides the user keys, Alice and Bob maintain their own wallet addresses, respectively.

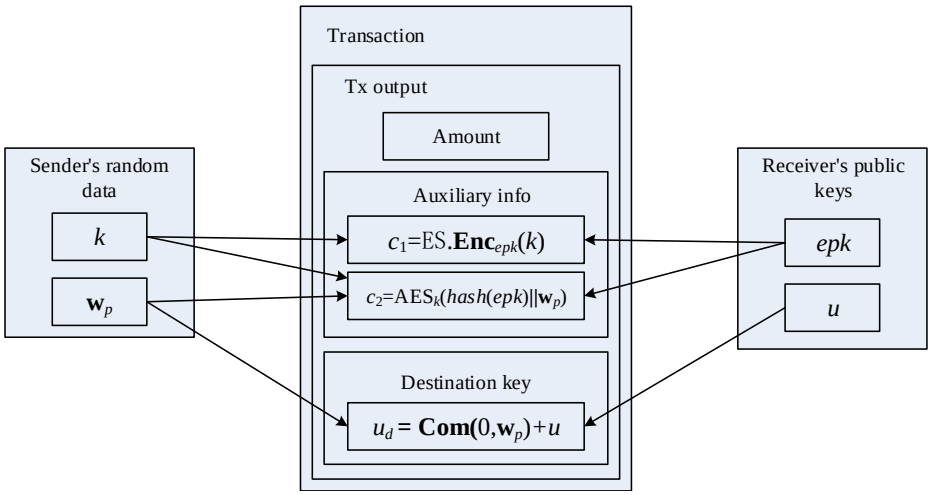

Fig. 1. DKeyGen procedure

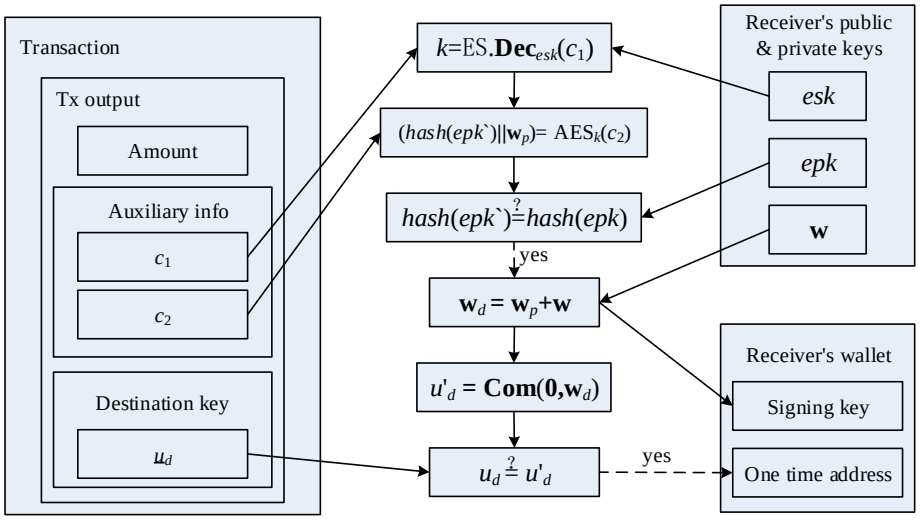

Fig. 2. DKeyRec procedure

Assume that the destination address u_{Bj} and its signing key $\mathbf{w}_{Bj}$ are in Alice's wallet, and she wants to send coins of this address to Bob. Alice will specify $\mu - 1$ foreign outputs ($\text{Output}_{B1}, \ldots, \text{Output}_{B(j-1)}, \text{Output}_{B(j+1)}, \ldots, \text{Output}_{B\mu}$) in which the amount is equivalent to that of Output_{Bj}. She proceeds to find Bob's public key $(epk_{\text{Bob}}, u_{\text{Bob}})$ and runs $\mathcal{KG}.\mathbf{DkeyGen}(pp, epk_{\text{Bob}}, u_{Bob})$ to generate the destination key u_{Cj} and its auxiliary information c_1, c_2 for Bob (see Fig. 1). She then pushes 1) Tx input including $(\text{Output}_{Bi})_{i=1}^{\mu}$ and the amount she sends to Bob, 2) the destination key u_{Cj} and auxiliary information c_1, c_2 she generated for Bob, 3) all previous transactions with output $\{\text{Output}_{Bi}\}_{i=1}^{\mu}$, into the hash function to obtain a hash digest, μ, of the transaction. Subsequently, she runs $\sigma \leftarrow \mathcal{LRS}.\mathbf{Sign}(pp, \mathbf{w}_{Bj}, \mu, event, u_{B1}, \ldots, u_{B\mu})$ to sign the hash digest, where u_{Bi} is the destination key of Output_{Bi}. Finally she broadcasts the transaction.

Bob checks all passing transactions. For each transaction, he extracts the destination key and auxiliary information (u_d, c_1, c_2), and runs the procedure $\mathcal{KG}.\mathbf{DKeyRec}(pp, epk_{Bob}, esk_{Bob}, u_{Bob}, \mathbf{w}_{Bob}, (u_d, c_1, c_2))$. If this transaction is the one that Alice sent to Bob, the foregoing procedure will return the signing key $\mathbf{w}_{Cj}$ for the destination key $u_d = u_{Cj}$. If this happens, Bob accepts this transaction and records $\mathbf{w}_{Cj}$, u_d into his wallet. Bob can later spend the coin stored in the destination address u_d because he has the signing key $\mathbf{w}_{Cj}$.

The standard transaction is also briefly depicted in Fig. 3.

7 Conclusions and Future Works

In this paper, using the techniques in [14] and [10], we constructed a linkable ring signature from ideal-lattices in which the size of a signature, on behalf of a ring with μ participants, is $O(\frac{n \log \mu}{\log(2n) - \log \tau})$. Based on the proposed signature scheme, we presented an anonymous post-quantum cryptocash system by following the major ideas of CryptoNote. In order to generate stealth addresses (verifying keys) and recover corresponding signing keys for the linkable ring signature, we provided a key-generation protocol as a subroutine of the cryptocash system. By combining all those techniques together, our cryptocash protocol obtains a new level anonymity comparing to the original Bitcoin system. Furthermore, the new designed cryptocash system has the potential to resist quantum attacks. We also notice that by using the technique in of Stern [39], confidential transactions is achievable based on lattices, and this will be one of our future works.

Recently, the unlinkability and untraceability of Monero were analyzed by [28] and [16]. Some of them were blamed on the abuses of users, *e.g.* signing a transaction on behalf of a ring with only 1 participant. Besides, there are still a few inherent weakness in Monero, *e.g.* for an overwhelming proportion of input addresses, a user can't find enough addresses with the same value to hide his real address, especially in the early time of the system. Next, we shall trace these problems and discuss what should be done to make our cryptocash system secure under these analyses. A full cryptocash system will be implement to test

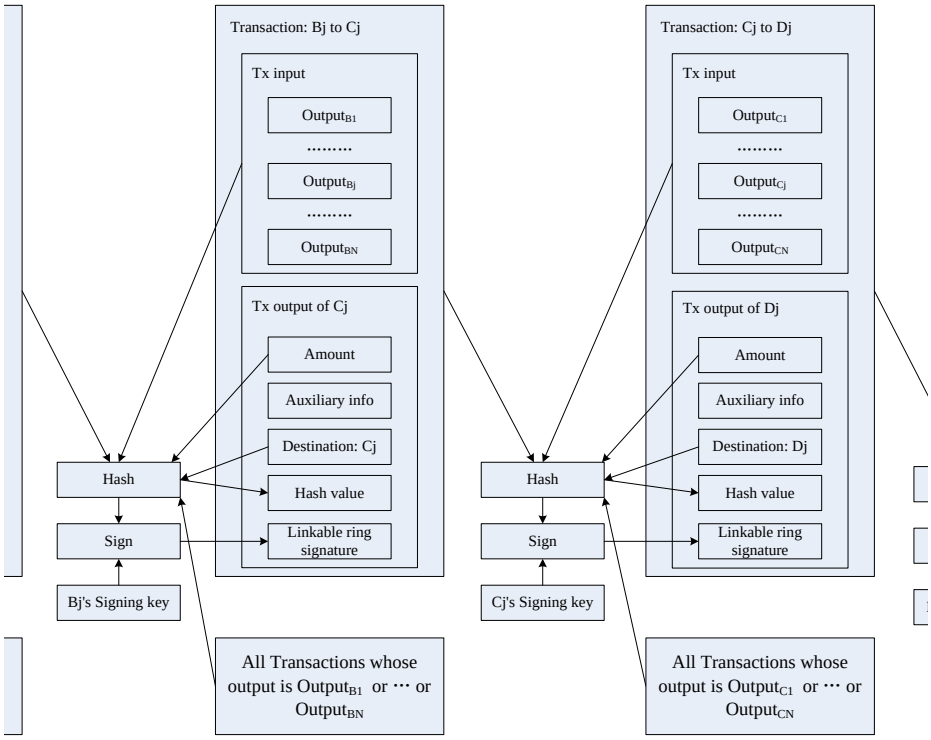

Fig. 3. Transaction chains

the communication and computation costs. And if possible, we would like to contribute our system to the cryptocash community for public usage.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61672550) and the National Key R&D Program of China (2017YFB0802503)

The authors are grateful to the anonymous reviewers for their valuable suggestions and comments on this paper.

References

1. Aharonov, D., Regev, O.: Lattice problems in NP ∩ coNP. J. ACM 52(5), 749–765 (sep 2005)
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Symposium on Theory of Computing—STOC 1996. pp. 99–108. ACM (1996)
3. Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to better—how to make Bitcoin a better currency. In: Financial Cryptography and Data Security—FC 2012. pp. 399–414. Springer Berlin Heidelberg (2012)
4. Baum, C., Lin, H., Oechsner, S.: Towards practical lattice-based one-time linkable ring signatures. In: Information and Communications Security. pp. 303–322. Springer International Publishing (2018)
5. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Advances in Cryptology—ASIACRYPT 2001. pp. 566–582. Springer Berlin Heidelberg (2001)
6. Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. In: Computer Security—ESORICS 2015. pp. 305–325. Springer International Publishing (2015)
7. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., Petit, C.: Short accountable ring signatures based on DDH. In: Computer Security—ESORICS 2015. pp. 243–265. Springer International Publishing (2015)
8. Cai, J.Y., Nerurkar, A.P.: An improved worst-case to average-case connection for lattice problems. In: Symposium on Foundations of Computer Science—FOCS 1997. pp. 468–477. IEEE (Oct 1997)
9. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal Gaussians. In: Advances in Cryptology—CRYPTO 2013. pp. 40–56. Springer Berlin Heidelberg (2013)
10. 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. Cryptology ePrint Archive, Report 2018/773 (2018), https://eprint.iacr.org/2018/773
11. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Public Key Cryptography—PKC 2007. pp. 181–200. Springer Berlin Heidelberg (2007)
12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Symposium on Theory of Computing—STOC 2008. pp. 197–206. ACM (2008), full version at https://eprint.iacr.org/2007/432

13. Goldreich, O., Goldwasser, S.: On the limits of non-approximability of lattice problems. In: Symposium on Theory of Computing—STOC 1998. pp. 1–9. ACM (1998)
14. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Advances in Cryptology—EUROCRYPT 2015. pp. 253–280. Springer Berlin Heidelberg (2015)
15. Halevi, S.: A sufficient condition for key-privacy. Cryptology ePrint Archive, Report 2005/005 (2005), http://eprint.iacr.org/2005/005
16. Kumar, A., Fischer, C., Tople, S., Saxena, P.: A traceability analysis of monero's blockchain. In: Computer Security—ESORICS 2017. pp. 153–173. Springer International Publishing (2017)
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: Advances in Cryptology—EUROCRYPT 2016. pp. 1–31. Springer Berlin Heidelberg (2016)
18. Liu, J.K., Au, M.H., Susilo, W., Zhou, J.: Linkable ring signature with unconditional anonymity. IEEE Transactions on Knowledge and Data Engineering 26(1), 157–165 (Jan 2014)
19. Liu, J.K., Wong, D.S.: Linkable ring signatures: Security models and new schemes. In: Computational Science and Its Applications—ICCSA 2005. pp. 614–623. Springer Berlin Heidelberg (2005)
20. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Advances in Cryptology—EUROCRYPT 2012. pp. 738–755. Springer Berlin Heidelberg (2012), Full version at https://eprint.iacr.org/2011/537
21. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Automata, Languages and Programming—ICALP 2006. pp. 144–155. Springer Berlin Heidelberg (2006)
22. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Advances in Cryptology—EUROCRYPT 2018. pp. 204–224. Springer International Publishing (2018)
23. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: Symposium on Foundations of Computer Science—FOCS 2002. pp. 356–365. IEEE (2002)
24. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. computational complexity 16(4), 365–411 (Dec 2007)
25. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM Journal on Computing 37(1), 267–302 (2007)
26. Micciancio, D., Regev, O.: Lattice-based Cryptography, pp. 147–191. Springer Berlin Heidelberg (2009)
27. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from Bitcoin. In: Symposium on Security and Privacy—SP 2013. pp. 397–411 (May 2013)
28. Miller, A., Möser, M., Lee, K., Narayanan, A.: An empirical analysis of linkability in the Monero blockchain. eprint arXiv:1704.04299 (2017), https://arxiv.org/abs/1704.04299
29. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2012), http://www.bitcoin.org/bitcoin.pdf
30. Noether, S., Mackenzie, A., the Monero Research Lab: Ring confidential transactions. Ledger 1(0), 1–18 (2016)
31. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the Bitcoin transaction graph. Future Internet 5(2), 237–250 (2013)

32. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (Jun 2000)
33. Reid, F., Harrigan, M.: An Analysis of Anonymity in the Bitcoin System, pp. 197–223. Springer New York (2013)
34. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Advances in Cryptology—ASIACRYPT 2001. pp. 552–565. Springer Berlin Heidelberg (2001)
35. Ron, D., Shamir, A.: Quantitative analysis of the full Bitcoin transaction graph. In: Financial Cryptography and Data Security—FC 2013. pp. 6–24. Springer Berlin Heidelberg (2013)
36. Saberhagen, N.v.: Cryptonote v2.0 (2013), https://cryptonote.org/whitepaper.pdf
37. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474 (May 2014)
38. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Advances in Cryptology—ASIACRYPT 2009. pp. 617–635. Springer Berlin Heidelberg (2009)
39. Stern, J.: A new paradigm for public key identification. IEEE Transactions on Information Theory 42(6), 1757–1768 (Nov 1996)
40. Sun, S.F., Au, M.H., Liu, J.K., Yuen, T.H.: RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero. In: Computer Security—ESORICS 2017. pp. 456–474. Springer International Publishing (2017)
41. Yang, R., Au, M.H., Lai, J., Xu, Q., Yu, Z.: Lattice-based techniques for accountable anonymity: Composition of abstract sterns protocols and weak PRF with efficient protocols from LWR. Cryptology ePrint Archive, Report 2017/781 (2017), https://eprint.iacr.org/2017/781

Appendix

A Short Linkable Ring Signature Based on ECDLP

Let μ be the size of the ring and $n = \log \mu$. Define $f_{j,1} = f_j = \ell_j e + a_j = \delta_{1\ell_j} e + a_j$, and $f_{j,0} = e - f_j = (1 - \ell_j)e - a_j = \delta_{0\ell_j} e - a_j$. For every $i \in [0, \mu)$ the product $\prod_{j=1}^{n} f_{j,i_j}$ is a polynomial in the indeterminate e of the form

$$p_i(e) = \prod_{j=1}^{n} (\delta_{i_j \ell_j} e) + \sum_{k=0}^{n-1} p_{i,k} e^k = \delta_{i\ell} e^n + \sum_{k=1}^{n-1} p_{i,k} e^k.$$

Here, $p_{i,k}$ is the coefficient of the kth degree term of the polynomial $p_i(e)$, and can be efficiently computed when $(a_j)_{j=1}^{n}$, i and ℓ are given.

The linkable ring signature based on ECDLP consists of five efficient procedures (**Setup**, **KGen**, **Sign**, **Vry**, **Link**).

Setup(1^λ): Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$. Let $G \in E$ be a point of prime order p, here $|p| = \lambda$ and let $\mathbb{G}$ be the prime order subgroup of E generated by G. Let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$, $\mathcal{H}' : \{0,1\}^* \to \mathbb{G}$ be two cryptographic hash functions. The output of this procedure is $pp = (\mathbb{G}, G, p, q, \mathcal{H}, \mathcal{H}')$.

KGen(pp): For the ith user, this procedure chooses the signing key $x_i \in \mathbb{Z}_p$ uniformly at random and computes the verifying key $Y_i = x_i G$. It outputs (x_i, Y_i) as the key pair of the ith user.

Sign$(x_\ell, \text{msg}, \mathit{event}, L)$: Let $L = (Y_0, Y_1, \ldots, Y_{\mu-1})$ be the ensemble of the ring. On input the message msg, the ℓ-th user's signature on behalf of L with event-id event is generated as follows

- Compute $H = \mathcal{H}'(\mathit{event})$, and $I = x_\ell H$.
- For j from 1 to n,
  - choose $r_j, a_j, s_j, t_j, \rho_k \leftarrow \mathbb{Z}_p$ at random.
  - compute $C_{\ell_j} = \ell_j H + r_j G$,
  - compute $C_{a_j} = a_j H + s_j G$,
  - compute $C_{b_j} = a_j \ell_j H + t_j G$,
- For k from 1 to $n - 1$
  - choose $\rho_k \leftarrow \mathbb{Z}_p$ at random,
  - compute $C_{d_k} = (\sum_{i=0}^{\mu-1} p_{i,k} Y_i) + \rho_k G$,
  - compute $C'_{d_k} = \rho_k H$.
- Let $\mathbf{a} = (C_{\ell_j}, C_{a_j}, C_{b_j}, C_{d_{j-1}}, C'_{d_{j-1}})_{j=1}^{n}$ and compute $e = \mathcal{H}(pp, \text{msg}, L, \mathbf{a}, I, \mathit{event})$.
- For j from 1 to n, compute
  - $f_j = e\ell_j + a_j$,
  - $z_{a_j} = e r_j + s_j$,
  - $z_{b_j} = (e - f_j) r_j + t_j$,
  - $z_d = e^n x_\ell - \sum_{k=0}^{n-1} e^k \rho_k$.
- Let $\mathbf{b} = (f_j, z_{a_j}, z_{b_j})_{j=1}^{n}$. Publish $\sigma = \{\mathbf{a}, \mathbf{b}, z_d, I\}$, the ring L, the event-id event, and the message msg.

Vry$(\text{msg}, event, L, \sigma)$:

- Compute $e = \mathcal{H}(pp, \text{msg}, L, \mathbf{a}, I, event)$, and $H = \mathcal{H}'(event)$.
- For j from 1 to n, consider the following equalities
  - $eC_{\ell_j} + C_{a_j} = f_j H + z_{a_j} G$,
  - $(e - f_j)C_{\ell_j} + C_{b_j} = z_{b_j} G$.

  If any one of them doesn't hold, output 0 and abort.
- If the equality $e^n I_\ell + \sum_{k=0}^{n-1}(-e^k)C'_{d_k} = z_d H$ doesn't hold, output 0 and abort.
- Inspect whether $\sum_{i=0}^{\mu-1}(\prod_{j=1}^{n} f_{j,i_j})Y_i + \sum_{k=0}^{n-1}(-e^k)C_{d_k} = z_d G$. If it is not, output 0 and abort; otherwise output 1.

Link(pp, σ_1, σ_2): For two accepting signatures $\sigma_1 = (\ldots, I_1)$ and $\sigma_2 = (\ldots, I_2)$ on the same event-id $event$, if $I_1 = I_2$, return 1 (linked) for concluding that they are generated by the same signer; otherwise, return 0 (unlinked).

Note that the Pedersen commitment of value 0 can act as a public key of our ECDLP-based linkable ring signature. As a result, the technique of RingCT [30] (later strengthened by Sun *et al.* [40]), which is adopted in Monero to hide the amount of a transaction, is trivially achievable in our settings. Using the above logarithmic size linkable ring signature to replace the linkable ring signature scheme in Monero, we can implement a more efficient Monero system.