# Identity-based Unidirectional Proxy Re-Encryption and Re-Signature in Standard Model: Lattice-based Constructions*

Priyanka Dutta†, Willy Susilo, Dung Hoang Duong, Joonsang Baek, and Partha Sarathi Roy
Institute of Cybersecurity and Cryptology
School of Computing and Information Technology
University of Wollongong
Northfields Avenue, Wollongong NSW 2522, Australia
{pdutta,wsusilo,hduong,baek,partha}@uow.edu.au

**Abstract**

Proxy re-encryption (PRE) securely enables the re-encryption of ciphertexts from one key to another, without relying on trusted parties, i.e., it offers delegation of decryption rights. PRE allows a semi-trusted third party termed as a "proxy" to securely divert encrypted files of a user (delegator) to another user (delegatee) without revealing any information about the underlying files to the proxy. Whereas, Proxy re-signature (PRS) allows a semi-trusted proxy to convert a signature of a user (delegatee) into a signature of another user (delegator) on the same message, but the proxy cannot produce new valid signature on new messages for either delegator or delegatee. To eliminate the necessity of having a costly certificate verification process, Green and Ateniese [18] introduced an identity-based PRE (IB-PRE) and Shao et al. [32] introduced an identity-based PRS (IB-PRS). The potential applicability of IB-PRE and IB-PRS leads to intensive research from its first instantiations. Unfortunately, till today, there is no unidirectional IB-PRE and IB-PRS secure in the standard model, which can withstand quantum attack. In this paper, we provide, for the first time, concrete constructions of unidirectional IB-PRE and IB-PRS which are secure in standard model based on the hardness of learning with error problem and small integer solution problem, respectively. Our technique is to use the novel trapdoor delegation technique of Micciancio and Peikert. The way we use trapdoor delegation technique may prove useful for functionalities other than PRE and PRS as well.

**Keywords**: Learning with error, Small integer solution, Proxy Re-Encryption, Proxy Re-Signature

## 1 Introduction

Proxy Re-encryption (PRE) allows a semi-trusted third party, called a proxy, to securely divert encrypted files of one user (delegator) to another user (delegatee). The proxy, however, cannot learn the underlying message $m$, and thus both parties' privacy can be maintained. This primitive (and its variants) have various applications ranging from encrypted email forwarding [7], securing distributed file systems [4], to digital rights management systems [35]. We notice a real-world file system employing a PRE scheme by Toshiba Corporation [28]. In addition application-driven purposes, various works have shown connections between re-encryption with other cryptographic primitives, such as program obfuscation [19, 12, 11] and fully-homomorphic encryption [10]. Thus studies along this line are both important and interesting for theory and practice. The other primitive, proxy re-signature (PRS), allows a semi-trusted proxy to transform the delegatee's signature on a message into the delegator's signature on the same message, but the proxy cannot produce new valid signature on new messages for either delegator

*Journal of Internet Services and Information Security (JISIS)*, volume: 10, number: 4 (November 2020), pp. 1-22
DOI: *10.22667/JISIS.2020.11.30.001*

*This is the full version of a paper that appeared in WISA 2020 [14].

†Corresponding author: Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia, Tel: +61-24221-5632

or delegatee. PRS is employed in various applications including managing group signatures, providing proof that a certain path in a graph is taken.

Blaze, Bleumer, and Strauss [7] introduced the concept of PRE and PRS. PRE and PRS are classified as unidirectional and bidirectional based on the direction of delegation. It is worth mentioning that the unidirectional constructions are much desirable because bidirectional construction easily implementable using a unidirectional one. Though the concept of PRE and PRS were initiated in [7], the first unidirectional PRE and PRS proposed by Ateniese et al. in [4] and [5], respectively, where desired properties of PRE and PRS are listed. Desired properties of PRE are as follows: *Non-interactivity* (*re-encryption key* can be generated by the delegator alone using public information of the delegatee; no trusted authority is needed); *Proxy transparency* (neither the delegator nor the delegatees are aware of the presence of a proxy); *Key optimality* (the size of $B$'s secret key remains constant, regardless of how many delegations he accepts); *Collusion resilience* (it is computationally infeasible for the coalition of the proxy and user $B$ to compute $A$'s secret key); *Non-transitivity* (it should be hard for the proxy to re-delegate the decryption right, namely to compute $rk_{A\to C}$ from $rk_{A\to B}$, $rk_{B\to C}$). In case of PRS, desired properties are: *Non-transitivity*; *Proxy transparency*; *Key optimality*; *Non-interactivity*; *Private Proxy* (re-signing key can be kept secret by the proxy). To achieve the aforementioned properties with improved security guarantee, there are elegant followup works that can be found in [9, 19, 24, 12, 11, 31, 13, 23]. For the quantum-safe version of PRE, Gentry [16] mentioned the feasibility of unidirectional PRE through a fully homomorphic encryption scheme (FHE). However, FHE costs huge computation. Further development of lattice-based PRE can be found in [37, 22, 11, 29, 15]. Recently, Fan et al. [15] proposed lattice-based PRS.

Certificate management problem is a crucial issue in the PKI based schemes. This crucial issue was addressed by Green et al. [18] in the area of PRE and Shao et al. [31, 32] for PRS. For lattice-based construction, Singh et al. [33] proposed a bidirectional identity-based PRE. However, it is required to use secret key of both delegator and delegatee to generate re-encryption key, which lacks one of the fundamental properties of PRE. Further, they proposed unidirectional identity-based PRE [34], termed as IB-uPRE, secure in the random oracle model. However, the size of the re-encrypted ciphertext blows up than the original encrypted one. Moreover, the schemes encrypt the message bit by bit. Later, there are some further attempts to construct lattice-based identity-based PRE, which are flawed[1] [21, 38]. On the other hand, for lattice-based identity-based PRS, Tian [36] proposed a bidirectional construction which is secure in the random oracle model. Unfortunately, there is no post-quantum secure unidirectional identity-based PRS (IB-uPRS) secure even in the random oracle model.

**Our Contributions:** Constructions of post-quantum secure IB-uPRE and IB-uPRS, in the standard model, are interesting open research problems. In this paper, we resolve these daunting tasks by constructing concrete schemes based on the hardness of *learning with error* (LWE) problem and *small integer solution* (SIS) problem. The proposed IB-uPRE and IB-uPRS enjoy the properties like non-interactivity, proxy transparency, key optimality, and non-transitivity. Moreover, the proposed IB-uPRE is capable of encrypting a multi-bit message in one go. To construct the IB-uPRE and IB-uPRS, we start with the construction of the identity-based encryption scheme by Agrawal et al. [1]. In non-interactive IB-uPRE and IB-uPRS, it is required to construct re-encryption and re-signing key by the delegator alone. One of the feasible ways to adopt the non-interactive feature is to provide a trapdoor to the delegator as a secret key. But, this technique is not supported by the design of [1]. In [1], the trapdoor is the master secret key and the secret key of the user is sampled by the master secret key. We first trace the design of selective IBE and IBS, where the secret key of a user is also a trapdoor, by using the trapdoor delegation technique of [25]. Here, the secret key of a user is a tuple of trapdoor, where one is used for decryption

[1] In [21], authors claimed to prove IND-ID-CPA, but provide the proof for IND-CPA. In [38], authors assumed a universally known entity ($\mathbf{G}$ matrix; see section 2.1) as a secret entity.

or signing and another one is used for re-encryption or re-signing key (ReKey) generation. ReKey is generated as in [22, 15] with a trick to resists proxy to get any secret information of delegator. The underlying IBE and IBS of the proposed IB-uPRE and IB-uPRS may prove useful to design expressive cryptographic primitives other than IB-PRE and IB-PRS as well.

**Overview and Techniques:** For IB-uPRE scheme, we consider identities as elements from $\mathbb{Z}_q^n$. In **SetUp** phase, we choose uniformly random matrix $\bar{\mathbf{A}}$ from $\mathbb{Z}_q^{n\times\bar{m}}$ and a random "short" matrix $\mathbf{R}$ from the Gaussian distribution $D_{\mathbb{Z},r}^{\bar{m}\times nk}$. Set $\bar{\mathbf{A}}' = -\bar{\mathbf{A}}\mathbf{R}$. So, for $\left[\begin{array}{c|c} \bar{\mathbf{A}} & -\bar{\mathbf{A}}\mathbf{R} \end{array}\right]$, $\mathbf{R}$ is a trapdoor with $\mathbf{0}$ tag. We also choose four invertible matrices $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3$, and $\mathbf{H}_4$ from $\mathbb{Z}_q^{n\times n}$ and two random matrices $\mathbf{A}_1, \mathbf{A}_2$ from $\mathbb{Z}_q^{n\times nk}$. The setup algorithm outputs $\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2$ together with the invertible matrices as Public parameters and that trapdoor $\mathbf{R}$ as the Master secret Key. To compute the secret key for an identity $id_i$, we first construct $\tilde{\mathbf{A}}_i = \left[\begin{array}{c|c} \bar{\mathbf{A}} & -\bar{\mathbf{A}}\mathbf{R} + \mathbf{H}_{id_i}\mathbf{G} \end{array}\right]$ (where $\mathbf{H}_{id_i}$ is output of FRD[1], $\mathbf{G}$ is the gadget matrix [25]), for which $\mathbf{R}$ is also a trapdoor with invertible tag $\mathbf{H}_{id_i}$. We construct $\mathbf{A}_{i1}, \mathbf{A}_{i2}$ as $\mathbf{A}_1 + \mathbf{H}_3\mathbf{H}_{id_i}\mathbf{G}$, and $\mathbf{A}_2 + \mathbf{H}_4\mathbf{H}_{id_i}\mathbf{G}$ respectively. Finally, we use the novel delegation technique for $\left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} \end{array}\right]$, $\left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i2} \end{array}\right]$ to get trapdoors $\mathbf{R}_{i1}, \mathbf{R}_{i2}$ respectively. The **Extract** algorithm outputs $\left[\begin{array}{c|c} \mathbf{R}_{i1} & \mathbf{R}_{i2} \end{array}\right]$, as the secret Key for the identity $id_i$. In Our scheme, we use $\mathbf{R}_{i1}$ to compute Re-encryption key from $id_i$ to any users, and $\mathbf{R}_{i2}$ for decryption. For Encryption, and Decryption, we use the method of Micciancio and Peikert Encryption Scheme [25]. To compute the re-encrypted ciphertext, we simply multiply the fresh ciphertext with Re-encryption Key. This re-encrypted ciphertext can be decrypted with similar method as for fresh ciphertext. Such technique enables the properties like proxy transparency and key optimality. For the selective security, let $id^*$ be the target identity given by the Adversary. During **SetUp** phase, we first set $\bar{\mathbf{A}}' = -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id^*}\mathbf{G}$. For which, for any user other than the target identity $id^*$, $\tilde{\mathbf{A}}_i$ will be $\left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} + (\mathbf{H}_{id_i} - \mathbf{H}_{id^*})\mathbf{G} \end{array}\right]$. Since, $(\mathbf{H}_{id_i} - \mathbf{H}_{id^*})$ is invertible for all $id \neq id^*$, we can delegate $sk_{id}$ using $\mathbf{R}$. We also choose $\mathbf{R}_{i^*1}, \mathbf{R}_{i^*2} \leftarrow D_{\mathbb{Z},r}^{m\times nk}$, which we use for reduction. We set $\mathbf{A}_1' = -\left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} \end{array}\right] \cdot \mathbf{R}_{i^*1}$ and $\mathbf{A}_2' = -\left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} \end{array}\right] \cdot \mathbf{R}_{i^*2}$. Construct $\mathbf{A}_1, \mathbf{A}_2$ as $\mathbf{A}_1' - \mathbf{H}_3\mathbf{H}_{id^*}\mathbf{G}$ and $\mathbf{A}_2' - \mathbf{H}_4\mathbf{H}_{id^*}\mathbf{G}$, respectively. So for $id^*$, $\tilde{\mathbf{A}}_{i^*}$ becomes $\left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} \end{array}\right]$, and $\mathbf{A}_{i^*} = \left[\begin{array}{c|c|c} \tilde{\mathbf{A}}_{i^*} & -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} & -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2} \end{array}\right]$. Thus, we can embed an LWE challenge at $id^*$.

For the IB-uPRS scheme, we employ the concept of tag-based signature scheme [8, 15], where each signature carries a Sign-tag that can be chosen uniformly during signing from a suitable Sign-tag set. Here, we do almost the same as the IB-uPRE scheme with the following exceptions: we choose $l+1$ elements $\mathbf{a}, \mathbf{b}_1, \cdots, \mathbf{b}_l$ from $\mathbb{Z}_q^n$ as public parameters; we choose an invertible matrix $\mathbf{H}_1$ from $\mathbb{Z}_q^{n\times n}$. For the signing key of identity $id_i$, we do exactly the same as the IB-uPRE scheme, except we compute $\mathbf{R}_{i2}$ with $\mathbf{0}$ tag. Here, we use $\mathbf{R}_{i1}$ to compute the Re-signing key from any users to $id_i$, and $\mathbf{R}_{i2}$ for signing the message. During the signing of a $l$-bit message $\mathbf{m}$, we first choose a Sign-tag $\mathbf{t}$ from the Sign-tag space $\mathscr{T}$ and compute $\mathbf{H}_{\mathbf{t}}$. Then we construct the signing matrix $\mathbf{A}_{id_i,\mathbf{t}} = \left[\begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} & \mathbf{A}_{i2} + \mathbf{H}_{\mathbf{t}} \end{array}\right] = \left[\begin{array}{c|c|c} \tilde{\mathbf{A}}_i & -\tilde{\mathbf{A}}_i\mathbf{R}_{i1} + \mathbf{H}_1\mathbf{G} & -\tilde{\mathbf{A}}_i\mathbf{R}_{i2} + \mathbf{H}_{\mathbf{t}} \end{array}\right]$. We sample a vector $\mathbf{e}_1 \in \mathscr{D}_{\mathbb{Z}^{nk},s}$. Finally, we sample the vector $(\mathbf{e}_0, \mathbf{e}_2) \in \mathbb{Z}^m \times \mathbb{Z}^{nk}$ for the cosets obtained from $\mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i - (-\tilde{\mathbf{A}}_i\mathbf{R}_{i1} + \mathbf{H}_1\mathbf{G})\mathbf{e}_1$, where $m_i$ is the $i^{th}$ bit of $\mathbf{m}$; we use $\mathbf{Sample}^{\mathscr{O}}$ for the matrix $\left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i2} + \mathbf{H}_{\mathbf{t}} \end{array}\right]$ with the generalized trapdoor $\mathbf{R}_{i2}$ and tag $\mathbf{H}_{\mathbf{t}}$. For which, it holds that $\mathbf{A}_{id_i,t} \cdot (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$. We output the signature $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ with the corresponding Sign-tag $\mathbf{t}$. For verification, we need to check the norm of $\mathbf{e}$, and $\mathbf{A}_{id_i,t} \cdot \mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$ or not. To compute the re-signature, we simply multiply the fresh signature with Re-signing Key. Verification of signature and re-signature follow the same algorithm, which enables the properties like proxy transparency and key optimality. Now for the security reduction, let the target identity be $id^*$ and target Sign-tag be $\mathbf{t}^*$, given by the Adversary. Here we will do same as IB-uPRE, except we construct $\mathbf{A}_2$ as $\mathbf{A}_2' - \mathbf{H}_{id^*}\mathbf{G} - \mathbf{H}_{\mathbf{t}^*}\mathbf{G}$. Since the trapdoor for computing signatures dose not vanish, for the signing

query and re-signing query the reduction can generate signatures from the same distribution as long as the Sign-tag is not $\mathbf{t}^*$. At the end of experiment, adversary outputs a forgery tuple $(id^*, m^*, \mathbf{e}^*, \mathbf{t}^*)$, which vanishes the trapdoors in target user. Then the reduction can derive an SIS solution from the forgery tuple.

**Improvements from the proceedings version [14]:** This version includes the following new results:

1. We newly propose a lattice-based construction of IB-uPRS in Section 4 and add required prerequisites in Section 2.
2. We add a details proof of Theorem 16.

# 2 Preliminaries

We denote the real numbers and the integers by $\mathbb{R}, \mathbb{Z}$, respectively. We denote column-vectors by lowercase bold letters (e.g. $\mathbf{b}$), so row-vectors are represented via transposition (e.g. $\mathbf{b}^t$). Matrices are denoted by upper-case bold letters and treat a matrix $\mathbf{X}$ interchangeably with its ordered set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ of column vectors. We use $\mathbf{I}$ for the identity matrix and $\mathbf{0}$ for the zero matrix, where the dimension will be clear from context. We use $[*|*]$ to denote the concatenation of vectors or matrices. A negligible function, denoted generically by $\mathsf{negl}(n)$. We say that a probability is overwhelming if it is $1 - \mathsf{negl}(n)$. The *statistical distance* between two distributions $\mathbf{X}$ and $\mathbf{Y}$ over a countable domain $\Omega$ defined as $\frac{1}{2}\sum_{w\in\Omega} |\mathsf{Pr}[\mathbf{X} = w] - \mathsf{Pr}[\mathbf{Y} = w]|$. We say that a distribution over $\Omega$ is $\varepsilon$-far if its statistical distance from the uniform distribution is at most $\varepsilon$. Throughout the paper, $r = \omega(\sqrt{\log n})$ represents a fixed function which will be approximated by $\sqrt{\ln(2n/\varepsilon)/\pi}$.

## 2.1 Lattices

A *lattice* $\Lambda$ is a discrete additive subgroup of $\mathbb{R}^m$. Specially, a lattice $\Lambda$ in $\mathbb{R}^m$ with basis $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_n] \in \mathbb{R}^{m\times n}$, where each $\mathbf{b}_i$ is written in column form, is defined as $\Lambda := \{\sum_{i=1}^n \mathbf{b}_i x_i | x_i \in \mathbb{Z}\ \forall i = 1, \ldots, n\} \subseteq \mathbb{R}^m$. We call $n$ the rank of $\Lambda$ and if $n = m$ we say that $\Lambda$ is a full rank lattice. The dual lattice $\Lambda^*$ is the set of all vectors $\mathbf{y} \in \mathbb{R}^m$ satisfying $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$ for all vectors $\mathbf{x} \in \Lambda$. If $\mathbf{B}$ is a basis of an arbitrary lattice $\Lambda$, then $\mathbf{B}^* = \mathbf{B}(\mathbf{B}^t\mathbf{B})^{-1}$ is a basis for $\Lambda^*$. For a full-rank lattice, $\mathbf{B}^* = \mathbf{B}^{-t}$.

In this paper, we mainly consider full rank lattices containing $q\mathbb{Z}^m$, called $q$-ary lattices, defined as the following, for a given matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$

$$\Lambda^{\perp}(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = 0 \mod q\}.$$
$$\Lambda(\mathbf{A}^t) := \{\mathbf{z} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^n \ s.t.\ \mathbf{z} = \mathbf{A}^t\mathbf{s} \mod q\}.$$
$$\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{u} \mod q\} = \Lambda^{\perp}(\mathbf{A}) + \mathbf{x} \ for\ \mathbf{x} \in \Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}).$$

Note that, $\Lambda^{\perp}(\mathbf{A})$ and $\Lambda(\mathbf{A}^t)$ are dual lattices, up to a $q$ scaling factor: $q\Lambda^{\perp}(\mathbf{A})^* = \Lambda(\mathbf{A}^t)$, and vice-versa. Sometimes we consider the non-integral, 1-*ary* lattice $\frac{1}{q}\Lambda(\mathbf{A}^t) = \Lambda^{\perp}(\mathbf{A})^* \supseteq \mathbb{Z}^m$.

**Gaussian on Lattices:** The $n$-dimensional Gaussian function on $\mathbb{R}^n$ centered at 0, is defined as $\rho(\mathbf{x}) = \exp(-\pi \cdot \|\mathbf{x}\|^2),\ \forall\, \mathbf{x} \in \mathbb{R}^n$. For any matrix $\mathbf{B}$, we define a density function of a Gaussian distribution for $\mathbf{x} \in span(\mathbf{B})$ and for $\Sigma = \mathbf{B}\mathbf{B}^t \geq 0$ as $\rho_{\sqrt{\Sigma}} = \rho(\mathbf{B}^+\mathbf{x}) = \exp(-\pi \cdot \mathbf{x}^t\Sigma^+\mathbf{x})$.

Normalizing the above expression by its total measure over $span(\Sigma)$, we obtain a probability density function of the continuous Gaussian distribution $D_{\sqrt{\Sigma}}$. The covariance matrix of this distribution is $\frac{\Sigma}{2\pi}$, we ignore the $\frac{1}{2\pi}$ factor and refer to $\Sigma$ as the covariance matrix of $D_{\sqrt{\Sigma}}$.

The continuous Gaussian distribution $D_{\sqrt{\Sigma}}$ can be discretized to a lattice (or to the "shift" of the lattice) as follows: for $\Lambda \subset \mathbb{R}^n,\ \mathbf{c} \in \mathbb{R}^n$ and positive semi-definite $\Sigma > 0$ such that $(\Sigma + \mathbf{c}) \cap span(\Sigma)$ is nonempty, the discrete Gaussian distribution is $D_{\Lambda+\mathbf{c},\sqrt{\Sigma}}(\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma}}(\mathbf{x})}{\rho_{\sqrt{\Sigma}}(\Lambda+\mathbf{c})}\ \forall \mathbf{x} \in \Lambda + \mathbf{c}$, where the denominator is merely a normalization factor.

One of the important invariants of a lattice is the *smoothing parameter* $\eta_\varepsilon$ (originally defined in [27]), defined as the following [25].

**Definition 1.** *Let $\Sigma$ be a positive semi-definite matrix i.e., $\Sigma \geq 0$ and a lattice $\Lambda \subset span(\Sigma)$,we say that $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$ if $\rho_{\sqrt{\Sigma^+}}(\Lambda^*) \leq 1+\varepsilon$.*

We will also use the following tail bound on discrete Gaussians.

**Lemma 2** ([6, Lemma 1.5])**.** *Let $\Lambda \subset \mathbb{R}^n$ be a lattice and $r \geq \eta_\varepsilon(\Lambda)$ for some $\varepsilon \in (0,1)$. For any $\mathbf{c} \in span(\Lambda)$, we have $\mathsf{Pr}[\|D_{\Lambda+\mathbf{c},r}\| \geq r\sqrt{n}] \leq 2^{-n} \cdot \frac{1+\varepsilon}{1-\varepsilon}$. If $\mathbf{c} = \mathbf{0}$ then the inequality holds for any $r > 0$, with $\varepsilon = 0$.*

Now we state some useful facts about subgaussian random variable and the singular value of a matrix. For any matrix $\mathbf{B} \in \mathbb{R}^{n\times k}$, there exists a singular value decomposition $\mathbf{B} = \mathbf{QDP}^t$, where $\mathbf{Q} \in \mathbb{R}^{n\times n}, \mathbf{P} \in \mathbb{R}^{k\times k}$ are orthogonal matrices, and $\mathbf{D} \in \mathbb{R}^{n\times k}$ is a diagonal matrix with non-negative entries $s_i(\mathbf{B}) \geq \mathbf{0}$ on the diagonal, in non-decreasing order. The $s_i(\mathbf{B})$ are called the singular values of $\mathbf{B}$ and $s_1(\mathbf{B}) = \max_{\mathbf{u}} \|\mathbf{Bu}\| = \max_{\mathbf{u}} \|\mathbf{B}^t\mathbf{u}\| \geq \|\mathbf{B}\|, \|\mathbf{B}^t\|$, where the maximum is taken over all the real unit vectors $\mathbf{u}$ according to the corresponding dimension.

**Definition 3** ([25])**.** *For $\delta \geq 0$, a random variable $\mathbf{X}$ is $\delta$-subgaussian with parameter $s > 0$ if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies $\mathbb{E}[\exp(2\pi t\mathbf{X})] \leq \exp(\delta).\exp(\pi s^2 t^2)$.*

**Lemma 4.** *Let $\mathbf{A} \in \mathbb{R}^{n\times m}$ be a $\delta$-subgaussian random matrix with parameter s. There exist a universal constant $C > 0$ such that for any $t \geq 0$, we have $s_1(\mathbf{A}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$ except with probability at most $2\exp(\delta)\exp(-\pi t^2)$.*

**Lemma 5** ([20, Theorem 3.3.16])**.** *Let $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}$ and $l = \min\{m,n\}$. The following inequalities hold for the decreasingly ordered singular values of $\mathbf{AB}$ : $s_i(\mathbf{AB}) \leq s_i(\mathbf{A})s_1(\mathbf{B})$ for $i = 1,\ldots,l$.*

**Hard problems on Lattices:** There are two lattice-based one-way functions associated with matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for $m = poly(n)$:

- $g_{\mathbf{A}}(\mathbf{e},\mathbf{s}) = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \mod q$ for $\mathbf{s} \in \mathbb{Z}_q^n$ and a Gaussian $\mathbf{e} \in \mathbb{Z}^m$ and $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{Ax} \mod q$, for $\mathbf{x} \in \mathbb{Z}^m$;
- The Learning With Errors (LWE) problem was introduced in [30]. The problem to invert $g_{\mathbf{A}}(\mathbf{e},\mathbf{s})$, where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,\alpha q}$ is known as search-LWE$_{q,n,m,\alpha}$ problem and is as hard as quantumly solving Shortest Independent Vector Problem (SIVP) on $n$-dimensional lattices. The decisional-LWE$_{q,n,m,\alpha}$ problem asks to distinguish the output of $g_{\mathbf{A}}$ from uniform.
- The Small Integer Solution (SIS) problem was first suggested to be hard on average by Ajtai [2] and then formalized by Micciancio and Regev [27]. Finding a non-zero short preimage $\mathbf{x}'$ such that $f_{\mathbf{A}}(\mathbf{x}') = \mathbf{0}$, with $\|\mathbf{x}'\| \leq \beta$, is an instantiation of the SIS$_{q,n,m,\beta}$ problem. It is known to be as hard as certain worst-case problems (e.g. SIVP) in standard lattices [3, 27, 17, 26].

**Trapdoors for Lattices:** Here, we briefly describe the main results of [25] and it's generalized version from [22]: the definition of $\mathbf{G}$-trapdoor, the algorithms **Invert**$^{\mathcal{O}}$, **Sample**$^{\mathcal{O}}$ and **DelTrap**$^{\mathcal{O}}$.

A $\mathbf{G}$-trapdoor is a transformation (represented by a matrix $\mathbf{R}$) from a public matrix $\mathbf{A}$ to a special matrix $\mathbf{G}$ which is called as gadget matrix. The formal definitions as follows:

**Definition 6** ([25])**.** *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$ be matrices with $m \geq w \geq n$. A $\mathbf{G}$-trapdoor for $\mathbf{A}$ is a matrix $\mathbf{R} \in \mathbb{Z}^{(m-w)\times w}$ such that $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix} = \mathbf{HG}$, for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$. We refer to $\mathbf{H}$ as the tag of the trapdoor.*

**Definition 7** ([22]). *The generalized version of a* **G***-trapdoor* :
*Let* $\mathbf{A} = \left[\begin{array}{c|c|c|c} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{k-1} \end{array}\right] \in \mathbb{Z}_q^{n\times m}$ *for* $k \geq 2$, *and* $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times \bar{m}}, \mathbf{A}_1, \ldots, \mathbf{A}_{k-1} \in \mathbb{Z}_q^{n\times w}$ *with* $\bar{m} \geq w \geq n$ *and* $m = \bar{m} + (k-1)\cdot w$ (*typically*, $w = n\lceil \log q \rceil$). *A* **G***-trapdoor for* $\mathbf{A}$ *is a sequence of matrices* $\mathbf{R} = \left[\begin{array}{c|c|c|c} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{k-1} \end{array}\right] \in \mathbb{Z}_q^{\bar{m}\times(k-1)w}$ *such that* :

$$\left[\begin{array}{c|c|c|c} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{k-1} \end{array}\right] \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{k-1} \\ \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix} = \left[\begin{array}{c|c|c|c} \mathbf{H}_1\mathbf{G} & \mathbf{H}_2\mathbf{G} & \cdots & \mathbf{H}_{k-1}\mathbf{G} \end{array}\right],$$

*for invertible matrices* $\mathbf{H}_i \in \mathbb{Z}_q^{n\times n}$ *and a fixed* $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$.

**Invert**$^{\mathscr{O}}(\mathbf{R}, \mathbf{A}, \mathbf{b}, \mathbf{H}_i)$ [22]: On input a vector $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$, a matrix $\mathbf{A} = \left[\begin{array}{c|c|c|c} \mathbf{A}_0 & -\mathbf{A}_0\mathbf{R}_1 + \mathbf{H_1}\mathbf{G} & \cdots & -\mathbf{A}_0\mathbf{R}_{k-1} + \mathbf{H}_{k-1}\mathbf{G} \end{array}\right]$ and corresponding **G**- trapdoor $\mathbf{R} = \left[\begin{array}{c|c|c|c} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{k-1} \end{array}\right]$ with invertible tag $\mathbf{H}_i$, the algorithm computes

$$\mathbf{b}'^t = \mathbf{b}^t \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{k-1} \\ \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix}$$

and then run the inverting oracle $\mathscr{O}(\mathbf{b}')$ for $\mathbf{G}$ to get $(\mathbf{s}', \mathbf{e}')$. The algorithm outputs $\mathbf{s} = \mathbf{H}_i^{-1}\mathbf{s}'$ and $\mathbf{e} = \mathbf{b} - \mathbf{A}^t\mathbf{s}$. Note that, **Invert**$^{\mathscr{O}}$ produces correct output if $\mathbf{e} \in \mathscr{P}_{1/2}(q\cdot\mathbf{B}^{-t})$, where $\mathbf{B}$ is a basis of $\Lambda^{\perp}(\mathbf{G})$; cf. [25, Theorem 5.4].

**Sample**$^{\mathscr{O}}(\mathbf{R}, \mathbf{A}, \mathbf{H}, \mathbf{u}, s)$ [25]: On input $(\mathbf{R}, \mathbf{A}', \mathbf{H}, \mathbf{u}, s)$, the algorithm construct $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}' & -\mathbf{A}'\mathbf{R} + \mathbf{H}\mathbf{G} \end{array}\right]$, where $\mathbf{R}$ is the $\mathbf{G}$-trapdoor for matrix $\mathbf{A}$ with invertible tag $\mathbf{H}$ and $\mathbf{u} \in \mathbb{Z}_q^n$.The algorithm outputs, using an oracle $\mathscr{O}$ for Gaussian sampling over a desired coset $\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G})$, a vector drawn from a distribution within negligible statistical distance of $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),s}$, where $s \geq r\cdot\sqrt{s_1(\mathbf{R})^2+1}\sqrt{s_1(\Sigma_{\mathbf{G}})+2}$. To sample a Gaussian vector $\mathbf{x} \in \mathbb{Z}_q^m$ for $\mathbf{A} = \left[\begin{array}{c|c|c|c} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{k-1} \end{array}\right] \in \mathbb{Z}_q^{n\times m}$ with the generalized trapdoor $\mathbf{R} = \left[\begin{array}{c|c|c|c} \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_{k-1} \end{array}\right]$ and $k-1$ invertible $\mathbf{H}_i$'s given a coset $\mathbf{u} \in \mathbb{Z}_q^n$, use generalized version of **Sample**$^{\mathscr{O}}$ from [22].

**DelTrap**$^{\mathscr{O}}$ $(\mathbf{A}' = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A}_1 \end{array}\right], \mathbf{R}, \mathbf{H}', s)$[25]: On input an oracle $\mathscr{O}$ for discrete Gaussian sampling over cosets of $\Lambda = \Lambda^{\perp}(\mathbf{A})$ with parameter $s \geq \eta_{\varepsilon}(\Lambda^{\perp}(\mathbf{A}))$, an extended matrix $\mathbf{A}'$ of $\mathbf{A}$, an invertible matrix $\mathbf{H}'$, the algorithm will sample (using $\mathscr{O}$) each column of $\mathbf{R}'$ independently from a discrete Gaussian with parameter $s$ over the appropriate coset of $\Lambda^{\perp}(\mathbf{A})$, so that $\mathbf{A}\mathbf{R}' = \mathbf{H}'\mathbf{G} - \mathbf{A}_1$. The algorithm outputs a trapdoor $\mathbf{R}'$ for $\mathbf{A}'$ with tag $\mathbf{H}'$.

## 2.2 Identity-Based Unidirectional Proxy Re-Encryption

**Definition 8** (Identity-Based Unidirectional Proxy ReEncryption (IB-uPRE) [18]). *A unidirectional Identity-Based Proxy Re-Encryption* (IB-uPRE) *scheme is a tuple of algorithms* (**SetUp**, **Extract**, **ReKeyGen**, **Enc**, **ReEnc**, **Dec**) :

- $(PP, msk) \longleftarrow$ **SetUp**$(1^n)$ : *On input the security parameter* $1^n$, *the* **SetUp** *algorithm outputs* $PP, msk$.
- $sk_{id} \longleftarrow$ **Extract**$(PP, msk, id)$ : *On input an identity id, public parameter PP, master secret key, output the secret key* $sk_{id}$ *for id*.

- $rk_{i\to j} \longleftarrow$ **ReKeyGen**$(PP, sk_{id_i}, id_i, id_j)$ : *On input a public parameter PP, secret key $sk_{id_i}$ of a delegator i, and $id_i, id_j$, output a unidirectional re-encryption key $rk_{i\to j}$.*
- $ct \longleftarrow$ **Enc**$(PP, id, m)$ : *On input an identity id, public parameter PP and a plaintext $m \in \mathscr{M}$, output a ciphertext ct under the specified identity id.*
- $ct' \longleftarrow$ **ReEnc**$(PP, rk_{i\to j}, ct)$ : *On input a ciphertext ct under the identity i and a re-encryption key $rk_{i\to j}$, output a ciphertext $ct'$ under the identity j.*
- $m \longleftarrow$ **Dec**$(PP, sk_{id_i}, ct)$ : *On input the ciphertext ct under the identity i and secret key $sk_{id_i}$ of i, the algorithm outputs a plaintext m or the error symbol $\perp$.*

An Identity-Based Proxy Re-Encryption scheme is called single-hop if a ciphertext can be re-encrypted only once. In a multi-hop setting proxy can apply further re-encryptions to already re-encrypted ciphertext.

**Definition 9** (Single-hop IB-uPRE Correctness)**.** *A single-hop* IB-uPRE *scheme* (**SetUp**, **Extract**, **ReKeyGen**, **Enc**, **ReEnc**, **Dec**) *decrypts correctly for the plaintext space $\mathscr{M}$ if* :

- *For all $sk_{id}$, output by* **Extract** *under id and for all $m \in \mathscr{M}$, it holds that* **Dec**$(PP, sk_{id}, \textbf{Enc}(PP, id, m)) = m$.
- *For any re-encryption key $rk_{i\to j}$, output by* **ReKeyGen**$(PP, sk_{id_i}, id_i, id_j)$ *and any* $ct =$ **Enc**$(PP, id_i, m)$*, it holds that* **Dec**$(PP, sk_{id_j}, \textbf{ReEnc}(PP, rk_{i\to j}, ct)) = m$.

**Security Game of Unidirectional Selective Identity-Based Proxy Re-Encryption Scheme against Chosen Plaintext Attack (IND-sID-CPA)**: To describe the security model we first classify all of the users into honest $(HU)$ and corrupted $(CU)$. In the honest case an adversary does not know secret key, whereas for a corrupted user the adversary has secret key. Let $\mathscr{A}$ be the PPT adversary and $\Pi =$ (**SetUp**, **Extract**, **ReKeyGen**, **Enc**, **ReEnc**, **Dec**) be an IB-uPRE scheme with a plaintext space $\mathscr{M}$ and a ciphertext space $\mathscr{C}$. Let $id^* (\in HU)$ be the target user. Security game is defined according to the following game $\mathsf{Exp}^{\mathsf{IND\text{-}sID\text{-}CPA}}_{\mathscr{A}}(1^n)$ :

1. **SetUp**: The challenger runs **SetUp**$(1^n)$ to get $(PP, msk)$ and give $PP$ to $\mathscr{A}$.
2. **Phase 1:** The adversary $\mathscr{A}$ may make queries polynomially many times in any order to the following oracles:
   - $\mathscr{O}^{\textbf{Extract}}$: an oracle that on input $id \in CU$, output $sk_{id}$; Otherwise, output $\perp$.
   - $\mathscr{O}^{\textbf{ReKeyGen}}$: an oracle that on input the identities of $i$-th and $j$-th users: if $id_i \in HU \setminus \{id^*\}$, $id_j \in HU$ or $id_i, id_j \in CU$ or $id_i \in CU, id_j \in HU$, output $rk_{i\to j}$; otherwise, output $\perp$.
   - $\mathscr{O}^{\textbf{ReEnc}}$: an oracle that on input the identities of $i, j$-th users and ciphertext of $i$-th user: if $id_i \in HU \setminus \{id^*\}$, $id_j \in HU$ or $id_i, id_j \in CU$ or $id_i \in CU, id_j \in HU$output re-encrypted ciphertext; otherwise, output $\perp$.
3. **Challenge**: $\mathscr{A}$ outputs two messages $m_0, m_1 \in \mathscr{M}$ and is given a challenge ciphertext $ct_b \longleftarrow$ **Enc**$(PP, id^*, m_b)$ for either $b = 0$ or $b = 1$.
4. **Phase 2:** After receiving the challenge ciphertext, $\mathscr{A}$ continues to have access to the $\mathscr{O}^{\textbf{Extract}}$, $\mathscr{O}^{\textbf{ReKeyGen}}$ and $\mathscr{O}^{\textbf{ReEnc}}$ oracle as in **Phase 1**.
5. $\mathscr{O}^{\textbf{Decision}}$: On input $b'$ from $\mathscr{A}$, this oracle outputs 1 if $b = b'$ and 0 otherwise.

The advantage of an adversary in the above experiment $\mathsf{Exp}_{\mathscr{A}}^{\mathsf{IND\text{-}sID\text{-}CPA}}(1^n)$ is defined as $|\mathsf{Pr}[b'=b]-\frac{1}{2}|$.

**Definition 10.** *An* IB-uPRE *scheme is* IND-sID-CPA *secure if all PPT adversaries $\mathscr{A}$ have at most a negligible advantage in experiment* $\mathsf{Exp}_{\mathscr{A}}^{\mathsf{IND\text{-}sID\text{-}CPA}}(1^n)$.

**Remark 11.** *In [18],* **ReKeyGen** *query is allowed from $id^*$ to HU to make the* IB-uPRE *collusion resilient (coalition of malicious proxy and delegetee to compute delegator's secret key). Here, we have blocked* **ReKeyGen** *query from $id^*$ to HU and the proposed* IB-uPRE *scheme is not claimed to be collusion resilient.*

## 2.3 Identity-Based Unidirectional Proxy Re-Signature with selectively chosen tag

**Definition 12** (Identity-Based Unidirectional Proxy Re-Signature (IB-uPRS))**.** *A unidirectional Identity-Based Proxy Re-Signature* (IB-uPRS) *scheme is a tuple of algorithms* (**SetUp**, **Extract**, **Sign**, **Verify**, **ReKeyGen**, **ReSign**) *with a plaintext space $\mathscr{M}$ and a Sign-tag space $\mathscr{T}$:*

- $(PP, msk) \longleftarrow$ **SetUp**$(1^n)$ : *On input the security parameter $1^n$, the* **SetUp** *algorithm outputs a key pair $(PP, msk)$.*
- $sk_{id} \longleftarrow$ **Extract**$(PP, msk, id)$ : *On input an identity id, public parameter and master key, it outputs the signing key $sk_{id}$ for id.*
- $e \longleftarrow$ **Sign**$(PP, sk_{id}, m, t)$ : *On input public parameter PP, signing key $sk_{id}$, it computes a signature $\sigma$ for message $m \in \mathscr{M}$ and Sign-tag $t \in \mathscr{T}$ under the specified identity id.*
- *accept or reject* $\longleftarrow$ **Verify**$(PP, id, e, m, t)$ : *On input public parameter PP, identity id, the signature $\sigma$ for $m, t$ under the identity id, the algorithm outputs accept if the signature $\sigma$ is valid. Otherwise, it outputs reject.*
- $rk_{i\to j} \longleftarrow$ **ReKeyGen**$(PP, id_i, id_j, sk_{id_j})$ : *On input public parameter PP, two identities $id_i, id_j$ and the signing key $sk_{id_j}$ of a delegator $id_j$, it outputs a unidirectional re-signing key $rk_{i\to j}$.*
- $\sigma' \longleftarrow$ **ReSign**$(PP, rk_{i\to j}, id_i, id_j, m, t, \sigma)$ : *If* **Verify**$(PP, id_i, \sigma, m, t) = accept$, *then using re-signing key $rk_{i\to j}$ computes a re-signature $\sigma'$ of the message m, with Sign-tag t under the identity $id_j$, otherwise outputs $\perp$.*

**Definition 13** (Single-hop IB-uPRS Correctness)**.** *Let* $\sigma =$ **Sign**$(PP, sk_{id_i}, id_i, m)$ *and* $\sigma' =$ **ReSign**$(PP, rk_{i\to j}, id_i, id_j, m, t, \sigma)$ *of the same message m, then it holds that* **Verify**$(PP, id_i, \sigma, m, t) = accept$ *and* **Verify**$(PP, id_j, \sigma', m, t) = accept$.

**Security Game of Unidirectional Selective Identity-Based Proxy Re-Signature against Adaptive Chosen-Message Attack (EU-sID-CMA)** [32]: To describe the security model we first classify all of the users into honest user $(HU)$ and corrupted user $(CU)$. In the honest case an adversary does not know signing key, whereas for a corrupted user the adversary has the signing key. Let $\mathscr{A}$ be the PPT adversary and $\Pi =$ (**SetUp**, **Extract**, **Sign**, **Verify**, **ReKeyGen**, **Resign**) be an IB-uPRS scheme with a plaintext space $\mathscr{M}$ and a Sign-tag space $\mathscr{T}$. Let $id^* \in HU$ be the challenge identity and $t^* \in \mathscr{T}$ be the challenge Sign-tag, send by the adversary at the begining of the game. Security game is defined according to the following game played between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.

1. **SetUp**: The challenger $\mathscr{C}$ runs the $\mathbf{SetUp}(1^n)$ algorithm to get $(PP, msk)$, and then gives $PP$ to $\mathscr{A}$.

2. **Query Phase:** The adversary $\mathscr{A}$ may make queries polynomially many times in any order to the following oracles:

   - $\mathscr{O}^{\mathbf{Extract}}$: an oracle that on input $id \in CU$, output the signing key $sk_{id}$; Otherwise, output $\perp$.
   - $\mathscr{O}^{\mathbf{Sign}}$: an oracle that on input a message $m$, a Sign-tag $t \neq t^*$ under $id \in HU$, challenger runs the **Sign** algorithm to get a signature $\sigma$ and gives $\sigma$ to $\mathscr{A}$. It assumes that, for corrupted users, adversary can sign message by himself. If Sign-tag $t = t^*$, then challenger will output $\perp$.
   - $\mathscr{O}^{\mathbf{ReKeyGen}}$: an oracle that on input the identities of $i$-th and $j$-th users: if $id_i \in HU$, $id_j \in HU \setminus \{id^*\}$ or $id_i, id_j \in CU$ or $id_i \in HU, id_j \in CU$, output the Re-signing Key $rk_{i \to j}$; otherwise, output $\perp$.
   - $\mathscr{O}^{\mathbf{ReSign}}$: an oracle that on input the identities of $i, j$-th users and a signature $e$ of the message $m$, with Sign-tag $t$ under the identity $id_i$: if $id_i \in HU$, $id_j \in HU$ or $id_i, id_j \in CU$ or $id_i \in HU, id_j \in CU$ output re-signature; otherwise, output $\perp$. If Sign-tag $t = t^*$, then challenger will output $\perp$.

3. **Forgery**: $\mathscr{A}$ outputs a signature $\sigma^*$ on message $m^*$ with Sign-tag $t^*$ under $id^*$. The adversary succeeds if the following situations all hold:

   (a) $\mathbf{Verify}(PP, id^*, \sigma^*, m^*, t^*) = accept$.

   (b) No extract query made on $id^*$ to $\mathscr{O}^{\mathbf{Extract}}$.

   (c) No sign query made on $m^*$ for any Sign-tag $t$ under $id \in HU$ to $\mathscr{O}^{\mathbf{Sign}}$.

   (d) No re-signature query made on $(\sigma, m^*, id_i, id_j)$ for any Sign-tag $t$, where $id_j \in HU$ to $\mathscr{O}^{\mathbf{ReSign}}$.

The advantage of an adversary in the above game is the probability that $\mathscr{A}$ succeeds the game.

**Definition 14.** *An* IB-uPRS *scheme is said to be existential unforgeable against adaptive chosen message and selective identity attacks if all PPT adversaries $\mathscr{A}$ have at most a negligible advantage in the above game.*

**Remark 15.** *In contrast of [32], we don't allow the query to $\mathscr{O}^{\mathbf{ReKeyGen}}$ from $id_i \in HU$ to $id^*$.*

# 3 Single-hop Identity-Based Unidirectional Proxy Re-Encryption Scheme (IB-uPRE)

## 3.1 Construction of Single-hop IB-uPRE

In this section, we present our construction of single-hop IB-uPRE. We set the parameters as the following.

- $\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ is a gadget matrix for large enough prime power $q = poly(n)$ and $k = O(\log q) = O(\log n)$, so there are efficient algorithms to invert $g_{\mathbf{G}}$ and to sample for $f_{\mathbf{G}}$.
- $\bar{m} = O(nk)$ and the Gaussian $\mathscr{D} = D_{\mathbb{Z},r}^{\bar{m} \times nk}$, so that $(\bar{\mathbf{A}}, \bar{\mathbf{A}}\mathbf{R})$ is negl(n)-far from uniform for $\bar{\mathbf{A}}$.
- the LWE error rate $\alpha$ for IB-uPRE should satisfy $1/\alpha = O(nk)^3 \cdot r^3$.

To start out, we first recall encoding techniques from [25, 1].

- **Message Encoding:** In the proposed construction, message space is $\mathscr{M} = \{0,1\}^{nk}$. $\mathscr{M}$ map bijectively to the cosets of $\Lambda/2\Lambda$ for $\Lambda = \Lambda(\mathbf{G}^t)$ by some function *encode* that is efficient to evaluate and invert. In particular, letting $\mathbf{E} \in \mathbb{Z}^{nk \times nk}$ be any basis of $\Lambda$, we can map $\mathbf{m} \in \{0,1\}^{nk}$ to $encode(\mathbf{m}) = \mathbf{Em} \in \mathbb{Z}^{nk}$ [25].
- **Encoding of Identity:** In the following construction, we use *full-rank difference* map (FRD) as in [1]. FRD: $\mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$; $id \mapsto \mathbf{H}_{id}$. We assume identities are non-zero elements in $\mathbb{Z}_q^n$. The set of identities can be expanded to $\{0,1\}^*$ by hashing identities into $\mathbb{Z}_q^n$ using a collision resistant hash. FRD satisfies the following properties: 1. $\forall$ *distinct* $id_1, id_2 \in \mathbb{Z}_q^n$, the matrix $\mathbf{H}_{id_1} - \mathbf{H}_{id_2} \in \mathbb{Z}_q^{n \times n}$ is full rank; 2. $\forall\ id \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$, the matrix $\mathbf{H}_{id} \in \mathbb{Z}_q^{n \times n}$ is full rank; 3. FRD is computable in polynomial time (in $n \log q$).

The proposed IB-uPRE consists of the following algorithms:

**SetUp**$(1^n)$ : On input a security parameter $n$, do:

1. Choose $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} \leftarrow \mathscr{D}$, and set $\bar{\mathbf{A}}' = -\bar{\mathbf{A}}\mathbf{R} \in \mathbb{Z}_q^{n \times nk}$.
2. Choose four invertible matrices $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4$ uniformly random from $\mathbb{Z}_q^{n \times n}$.
3. Choose two random matrices $\mathbf{A}_1, \mathbf{A}_2$ from $\mathbb{Z}_q^{n \times nk}$.
4. Output $PP = (\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4, \mathbf{G})$ and the master secret key is $msk = \mathbf{R}$.

**Extract**$(PP, msk, id)$ : On input a public parameter $PP$, master secret key $msk$ and the identity of $i$-th user $id_i$, do:

1. Construct $\tilde{\mathbf{A}}_i = \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}' + \mathbf{H}_{id_i}\mathbf{G}\ \right] = \left[\ \bar{\mathbf{A}} \ \middle|\ -\bar{\mathbf{A}}\mathbf{R} + \mathbf{H}_{id_i}\mathbf{G}\ \right] \in \mathbb{Z}_q^{n \times m}$, where $m = \bar{m} + nk$. So, $\mathbf{R}$ is a trapdoor of $\tilde{\mathbf{A}}_i$ with tag $\mathbf{H}_{id_i}$.
2. - Construct $\mathbf{A}_{i1} = \mathbf{A}_1 + \mathbf{H}_3\mathbf{H}_{id_i}\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ and set $\mathbf{A}'_{i1} = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i1}\ \right] \in \mathbb{Z}_q^{n \times (m+nk)}$.
   - Call the algorithm $\mathbf{DelTrap}^{\mathscr{O}}(\mathbf{A}'_{i1}, \mathbf{R}, \mathbf{H}_1, s)$ to get a trapdoor $\mathbf{R}_{i1} \in \mathbb{Z}^{m \times nk}$ for $\mathbf{A}'_{i1}$ with tag $\mathbf{H}_1 \in \mathbb{Z}_q^{n \times n}$, where $s \geq \eta_\varepsilon(\Lambda^\perp(\tilde{\mathbf{A}}_i))$, so that $\tilde{\mathbf{A}}_i\mathbf{R}_{i1} = \mathbf{H}_1\mathbf{G} - \mathbf{A}_{i1}$.
3. - Construct $\mathbf{A}_{i2} = \mathbf{A}_2 + \mathbf{H}_4\mathbf{H}_{id_i}\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ and set $\mathbf{A}'_{i2} = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i2}\ \right] \in \mathbb{Z}_q^{n \times (m+nk)}$.
   - Call the algorithm $\mathbf{DelTrap}^{\mathscr{O}}(\mathbf{A}'_{i2}, \mathbf{R}, \mathbf{H}_2, s)$ to get a trapdoor $\mathbf{R}_{i2} \in \mathbb{Z}^{m \times nk}$ for $\mathbf{A}'_{i2}$ with tag $\mathbf{H}_2 \in \mathbb{Z}_q^{n \times n}$, so that $\tilde{\mathbf{A}}_i\mathbf{R}_{i2} = \mathbf{H}_2\mathbf{G} - \mathbf{A}_{i2}$.

   Output the secret key as $sk_{id_i} = \left[\ \mathbf{R}_{i1} \ \middle|\ \mathbf{R}_{i2}\ \right] \in \mathbb{Z}^{m \times 2nk}$. Notice that,

$$\left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i1} \ \middle|\ \mathbf{A}_{i2}\ \right] \begin{bmatrix} \mathbf{R}_{i1} & \mathbf{R}_{i2} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \left[\ \mathbf{H}_1\mathbf{G} \ \middle|\ \mathbf{H}_2\mathbf{G}\ \right].$$

**Enc**$(PP, id_i, \mathbf{m} \in \{0,1\}^{nk})$ : On input a public parameter $PP$, the identity of $i$-th user $id_i$ and message $\mathbf{m} \in \{0,1\}^{nk}$, do:

1. Construct $\tilde{\mathbf{A}}_i = \left[\ \bar{\mathbf{A}} \ \middle|\ -\bar{\mathbf{A}}\mathbf{R} + \mathbf{H}_{id_i}\mathbf{G}\ \right] \in \mathbb{Z}_q^{n \times m}$.
2. Construct $\mathbf{A}_{i1}, \mathbf{A}_{i2}$ for $id_i$ same as in **Extract** algorithm and set $\mathbf{A}_i = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i1} \ \middle|\ \mathbf{A}_{i2}\ \right]$.
3. Choose a uniformly random $\mathbf{s} \leftarrow \mathbb{Z}_q^n$.
4. Sample error vectors $\bar{\mathbf{e}_0} \leftarrow D^{\bar{m}}_{\mathbb{Z},\alpha q}$ and $\mathbf{e}'_0, \mathbf{e}_1, \mathbf{e}_2 \leftarrow D^{nk}_{\mathbb{Z},s'}$, where $s'^2 = (\|\bar{\mathbf{e}}_0\|^2 + \bar{m}(\alpha q)^2)r^2$. Let the error vector $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2) \in \mathbb{Z}^{\bar{m}+nk} \times \mathbb{Z}^{nk} \times \mathbb{Z}^{nk}$, where $\mathbf{e}_0 = (\bar{\mathbf{e}}_0, \mathbf{e}'_0) \in \mathbb{Z}^{\bar{m}} \times \mathbb{Z}^{nk}$.
5. Compute $\mathbf{b}^t = (\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2) = 2(\mathbf{s}^t\mathbf{A}_i \mod q) + \mathbf{e}^t + (\mathbf{0}, \mathbf{0}, encode(\mathbf{m})^t) \mod 2q$, where the first zero vector has dimension $\bar{m} + nk$, the second has dimension $nk$ and $\mathbf{b}_0 = (\bar{\mathbf{b}}_0, \mathbf{b}'_0)$.
6. Output the ciphertext $ct = \mathbf{b} \in \mathbb{Z}_{2q}^{\bar{m}+3nk}$.

**Dec**$(PP, sk_{id_i}, ct)$ : On input a public parameter $PP$, the secret key of $i$-th user $sk_{id_i}$ and ciphertext $ct$, do:

1. If $ct$ has invalid form or $\mathbf{H}_{id_i} = \mathbf{0}$, output $\perp$. Otherwise,
   - Construct $\tilde{\mathbf{A}}_i = \left[ \begin{array}{c|c} \bar{\mathbf{A}} & -\bar{\mathbf{A}}\mathbf{R}+\mathbf{H}_{id_i}\mathbf{G} \end{array} \right] \in \mathbb{Z}_q^{n\times m}$.
   - Construct $\mathbf{A}_{i1}, \mathbf{A}_{i2}$ for $id_i$ as in **Extract** algorithm and set $\mathbf{A}_i = \left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} & \mathbf{A}_{i2} \end{array} \right]$.
2. Call $\mathbf{Invert}^{\mathcal{O}}(\left[ \begin{array}{c|c} \mathbf{R}_{i1} & \mathbf{R}_{i2} \end{array} \right], \mathbf{A}_i, \mathbf{b}, \mathbf{H}_2)$ to get $\mathbf{z} \in \mathbb{Z}_q^n$ and $\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2) \in \mathbb{Z}^{\bar{m}+nk} \times \mathbb{Z}^{nk} \times \mathbb{Z}^{nk}$, where $\mathbf{e}_0 = (\bar{\mathbf{e}}_0, \mathbf{e}'_0) \in \mathbb{Z}^{\bar{m}} \times \mathbb{Z}^{nk}$ for which $\mathbf{b}^t = \mathbf{z}^t\mathbf{A}_i + \mathbf{e}^t \mod q$. If the call to **Invert** fails for any reason, output $\perp$.
3. If $\|\bar{\mathbf{e}}_0\| \geq \alpha q\sqrt{\bar{m}}$ or $\|\mathbf{e}'_0\| \geq \alpha q\sqrt{2\bar{m}nk} \cdot r$ or $\|\mathbf{e}_j\| \geq \alpha q\sqrt{2\bar{m}nk} \cdot r$ for $j = 1, 2$, output $\perp$.
4. Let $\mathbf{V} = \mathbf{b} - \mathbf{e} \mod 2q$, parsed as $\mathbf{V} = (\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2) \in \mathbb{Z}_{2q}^{\bar{m}+nk} \times \mathbb{Z}_{2q}^{nk} \times \mathbb{Z}_{2q}^{nk}$, where $\mathbf{V}_0 = (\overline{\mathbf{V}}_0, \mathbf{V}'_0) \in \mathbb{Z}_{2q}^{\bar{m}} \times \mathbb{Z}_{2q}^{nk}$. If $\overline{\mathbf{V}}_0 \notin 2\Lambda(\bar{\mathbf{A}}^t)$, output $\perp$.
5. Output $encode^{-1}(\mathbf{V}^t \begin{bmatrix} \mathbf{R}_{i1} & \mathbf{R}_{i2} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mod 2q) \in \{0,1\}^{nk}$ if it exists, otherwise output $\perp$.

**ReKeyGen**$(PP, sk_{id_i}, id_i, id_j)$ : On input a public parameter $PP$, the secret key of $i$-th user $sk_{id_i}$ and identity of $j$-th user $id_j$, do:

1. Construct $\mathbf{A}_i = \left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} & \mathbf{A}_{i2} \end{array} \right]$, where $\tilde{\mathbf{A}}_i = \left[ \begin{array}{c|c} \bar{\mathbf{A}} & \bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G} \end{array} \right]$ and $\mathbf{A}_{i1}, \mathbf{A}_{i2}$ are same as in **Extract** algorithm .
2. Construct $\mathbf{A}_j = \left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_j & \mathbf{A}_{j1} & \mathbf{A}_{j2} \end{array} \right]$, where $\tilde{\mathbf{A}}_j = \left[ \begin{array}{c|c} \bar{\mathbf{A}} & \bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G} \end{array} \right]$ and $\mathbf{A}_{j1}, \mathbf{A}_{j2}$ are same as in **Extract** algorithm .
3. Using $\mathbf{Sample}^{\mathcal{O}}$ with trapdoor $\mathbf{R}_{i1}$(from the secret key of $i$th user ), with tag $\mathbf{H}_1$, we sample from the cosets which are formed with the column of the matrix $\bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G}$. After sampling $nk$ times we get an $(\bar{m}+2nk) \times nk$ matrix and parse it as three matrices $\mathbf{X}_{00} \in \mathbb{Z}^{\bar{m}\times nk}$, $\mathbf{X}_{10} \in \mathbb{Z}^{nk\times nk}$ and $\mathbf{X}_{20} \in \mathbb{Z}^{nk\times nk}$ matrices with Gaussian entries of parameter $s$. So, $\left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & -\tilde{\mathbf{A}}_i\mathbf{R}_{i1}+\mathbf{H}_1\mathbf{G} \end{array} \right] \begin{bmatrix} \mathbf{X}_{00} \\ \mathbf{X}_{10} \\ \mathbf{X}_{20} \end{bmatrix} = \bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G}$, *i.e.* $\left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} \end{array} \right] \begin{bmatrix} \mathbf{X}_{00} \\ \mathbf{X}_{10} \\ \mathbf{X}_{20} \end{bmatrix} = \bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G}$.
4. Continue sampling for the cosets obtained from the columns of the matrix $\mathbf{A}_{j1}$ from $A_j$. This time, we increase the Gaussian parameter of the resulting sampled matrix up to $s\sqrt{\bar{m}/2}$: $\left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & -\tilde{\mathbf{A}}_i\mathbf{R}_{i1}+\mathbf{H}_1\mathbf{G} \end{array} \right] \begin{bmatrix} \mathbf{X}_{01} \\ \mathbf{X}_{11} \\ \mathbf{X}_{21} \end{bmatrix} = \mathbf{A}_{j1}$ , *i.e.* $\left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} \end{array} \right] \begin{bmatrix} \mathbf{X}_{01} \\ \mathbf{X}_{11} \\ \mathbf{X}_{21} \end{bmatrix} = \mathbf{A}_{j1}$ .

   For the last sampling, to get a correct re-encryption, we will use the cosets which are formed with the column of the matrix $\mathbf{A}_{j2} + \tilde{\mathbf{A}}_i\mathbf{R}_{i2} - \mathbf{H}_2\mathbf{G}$ :

   $\left[ \begin{array}{c|c|c} \tilde{\mathbf{A}}_i & -\tilde{\mathbf{A}}_i\mathbf{R}_{i1}+\mathbf{H}_1\mathbf{G} \end{array} \right] \begin{bmatrix} \mathbf{X}_{02} \\ \mathbf{X}_{12} \\ \mathbf{X}_{22} \end{bmatrix} = \mathbf{A}_{j2} + \tilde{\mathbf{A}}_i\mathbf{R}_{i2} - \mathbf{H}_2\mathbf{G}$ , where $\mathbf{X}_{01}, \mathbf{X}_{02} \in \mathbb{Z}^{\bar{m}\times nk}$ and $\mathbf{X}_{11}, \mathbf{X}_{12}, \mathbf{X}_{21}, \mathbf{X}_{22} \in \mathbb{Z}^{nk\times nk}$ with entries distributed as Gaussian with parameter $s\sqrt{\bar{m}}$.

5. Output re-encryption key $rk_{i\to j} = \begin{bmatrix} \mathbf{I} & \mathbf{X}_{00} & \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{0} & \mathbf{X}_{10} & \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{0} & \mathbf{X}_{20} & \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{Z}^{(m+2nk)\times(m+2nk)}$,

which satisfies: $\mathbf{A}_i \cdot rk_{i\to j} = \mathbf{A}_j$.

**ReEnc**$(rk_{i\to j}, ct)$ : On input $rk_{i\to j}$ and $i$-th user's ciphertext $ct$, Compute:
$\mathbf{b}'^t = \mathbf{b}^t \cdot rk_{i\to j} = 2\mathbf{s}^t \begin{bmatrix} \tilde{\mathbf{A}}_j \mid \mathbf{A}_{j1} \mid \mathbf{A}_{j2} \end{bmatrix} + \tilde{\mathbf{e}}^t + (\mathbf{0}, \mathbf{0}, encode(\mathbf{m})^t)$, where $\tilde{\mathbf{e}} = (\tilde{\mathbf{e}}_0, \tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2)$, $\tilde{\mathbf{e}}_0 = (\tilde{\bar{\mathbf{e}}}_0, \tilde{\mathbf{e}}'_0)$ and $\tilde{\bar{\mathbf{e}}}_0 = \bar{\mathbf{e}}_0$, $\tilde{\mathbf{e}}'_0 = \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20}$, $\tilde{\mathbf{e}}_1 = \bar{\mathbf{e}}_0\mathbf{X}_{01} + \mathbf{e}'_0\mathbf{X}_{11} + \mathbf{e}_1\mathbf{X}_{21}$, $\tilde{\mathbf{e}}_2 = \bar{\mathbf{e}}_0\mathbf{X}_{02} + \mathbf{e}'_0\mathbf{X}_{12} + \mathbf{e}_1\mathbf{X}_{22} + \mathbf{e}_2$.
Then output $ct' = \mathbf{b}'$.

## 3.2 Correctness and Security

In this section, we analyze the correctness and security of the proposed scheme.

**Theorem 16** (Correctness)**.** *The* IB-uPRE *scheme with parameters proposed in Section 3.1 is correct.*
**Proof:** To show that the decryption algorithm outputs a correct plaintext, we will consider both original and re-encrypted ciphertext. Let $sk_{id_i} = \begin{bmatrix} \mathbf{R}_{i1} \mid \mathbf{R}_{i2} \end{bmatrix}$ and $sk_{id_j} = \begin{bmatrix} \mathbf{R}_{j1} \mid \mathbf{R}_{j2} \end{bmatrix}$ be the secret key for $i$-th and $j$-th user respectively in the IB-uPRE scheme.
From **ReKeyGen**$(PP, sk_{id_i}, id_i, id_j)$ algorithm, we get

$$rk_{i\to j} = \begin{bmatrix} \mathbf{I}_{\bar{m}\times\bar{m}} & \mathbf{X}_{00} & \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{0} & \mathbf{X}_{10} & \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{0} & \mathbf{X}_{20} & \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{nk\times nk} \end{bmatrix}.$$

Let $ct = \mathbf{b}$ be the ciphertext of plaintext $\mathbf{m} \in \{0,1\}^{nk}$ for $i$-th user and $ct' = \mathbf{b}' = (\mathbf{ReEnc}(PP, rk_{i\to j}, ct))$ be the re-encrypted ciphertext for the $j$-th user. Thus, we need to prove that $\mathbf{Dec}(PP, sk_{id_i}, ct) = \mathbf{Dec}(PP, sk_{id_j}, ct') = \mathbf{m}$.

The arguments for the original ciphertext follows from the Lemma 6.2 of [25].

Now for the re-encrypted ciphertext, we have
$\mathbf{b}'^t = \mathbf{b}^t \cdot rk_{i\to j} = 2\mathbf{s}^t \begin{bmatrix} \tilde{\mathbf{A}}_j \mid \mathbf{A}_{j1} \mid \mathbf{A}_{j2} \end{bmatrix} + \tilde{\mathbf{e}}^t + (\mathbf{0}, \mathbf{0}, encode(\mathbf{m})^t) \mod 2q$, where $\mathbf{b}^t = 2(\mathbf{s}^t\mathbf{A}_i \mod q) + \mathbf{e}^t + (\mathbf{0}, \mathbf{0}, encode(\mathbf{m})^t) \mod 2q$,
$\mathbf{e} = (\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$, $\mathbf{e}_0 = (\bar{\mathbf{e}}_0, \mathbf{e}'_0)$ and $\tilde{\mathbf{e}} = (\tilde{\mathbf{e}}_0, \tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2)$, $\tilde{\mathbf{e}}_0 = (\tilde{\bar{\mathbf{e}}}_0, \tilde{\mathbf{e}}'_0)$ and $\tilde{\bar{\mathbf{e}}}_0 = \bar{\mathbf{e}}_0$, $\tilde{\mathbf{e}}'_0 = \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20}$, $\tilde{\mathbf{e}}_1 = \bar{\mathbf{e}}_0\mathbf{X}_{01} + \mathbf{e}'_0\mathbf{X}_{11} + \mathbf{e}_1\mathbf{X}_{21}$, $\tilde{\mathbf{e}}_2 = \bar{\mathbf{e}}_0\mathbf{X}_{02} + \mathbf{e}'_0\mathbf{X}_{12} + \mathbf{e}_1\mathbf{X}_{22} + \mathbf{e}_2$.

In the decryption process, we multiply a re-encrypted ciphertext (and thus its error term) by $\begin{bmatrix} \mathbf{R}_{j1} & \mathbf{R}_{j2} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$. So, in order to get a correct output, we need to show that $\tilde{\mathbf{e}}^t \cdot \begin{bmatrix} \mathbf{R}_{j1} & \mathbf{R}_{j2} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathscr{P}_{1/2}(q \cdot \mathbf{B}^{-t})$, where $\mathbf{B}$ is a basis of $\Lambda^{\perp}(\mathbf{G})$. i.e., $\tilde{\mathbf{e}}_0\mathbf{R}_{j1} + \tilde{\mathbf{e}}_1$, $\tilde{\mathbf{e}}_0\mathbf{R}_{j2} + \tilde{\mathbf{e}}_2 \in \mathscr{P}_{1/2}(q \cdot \mathbf{B}^{-t})$.

Here, $\tilde{\mathbf{e}}_0\mathbf{R}_{j1} + \tilde{\mathbf{e}}_1 = (\tilde{\bar{\mathbf{e}}}_0, \tilde{\mathbf{e}}'_0)\mathbf{R}_{j1} + \tilde{\mathbf{e}}_1 =$

$$(\bar{\mathbf{e}}_0, \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20})\mathbf{R}_{j1} + \bar{\mathbf{e}}_0\mathbf{X}_{01} + \mathbf{e}'_0\mathbf{X}_{11} + \mathbf{e}_1\mathbf{X}_{21} \tag{1}$$

$\tilde{\mathbf{e}}_0\mathbf{R}_{j2} + \tilde{\mathbf{e}}_2 = (\tilde{\bar{\mathbf{e}}}_0, \tilde{\mathbf{e}}'_0)\mathbf{R}_{j2} + \tilde{\mathbf{e}}_2 =$

$$(\bar{\mathbf{e}}_0, \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20})\mathbf{R}_{j2} + \bar{\mathbf{e}}_0\mathbf{X}_{02} + \mathbf{e}'_0\mathbf{X}_{12} + \mathbf{e}_1\mathbf{X}_{22} + \mathbf{e}_2 \tag{2}$$

So, we have to estimate the upper bounds for the length of (1) and (2). Hence, it is required to estimate the resultant length of the Gaussian vectors $\bar{\mathbf{e}}_0, \mathbf{e}'_0, \mathbf{e}_1, \mathbf{e}_2$ after multiplication by the matrices. We analyze each term of (1), separately. The same arguments hold for (2).

According to the sampling algorithm, the parameter $s$ for each column of the $\mathbf{X}_{00}, \mathbf{X}_{10}$ and of $\mathbf{X}_{20}$ is as small as $\sqrt{s_1(\mathbf{R}_{i1})^2+1}\cdot\sqrt{s_1(\Sigma_{\mathbf{G}})+2}\cdot r$, where $\mathbf{R}_{i1}$ is the trapdoor that was used in the re-encryption key generation. By Lemma 2, we have $\|\bar{\mathbf{e}}_0\| < \alpha q\sqrt{\bar{m}}$, $\|\mathbf{e}'_0\| < \alpha q\sqrt{2\bar{m}nk}\cdot r$ and $\|\mathbf{e}_1\| < \alpha q\sqrt{2\bar{m}nk}\cdot r$. Now from Lemma 4, Lemma 5 and the fact that $s_1(\Sigma_{\mathbf{G}}) = 4$, we obtain $\|(\bar{\mathbf{e}}_0, \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20})\| < \alpha q\cdot 12\sqrt{6}\cdot\sqrt{\bar{m}}\cdot\sqrt{2\bar{m}nk}\cdot\sqrt{s_1(\mathbf{R}_{i1})^2+1}\cdot r^2$ and $\|(\bar{\mathbf{e}}_0, \bar{\mathbf{e}}_0\mathbf{X}_{00} + \mathbf{e}'_0\mathbf{X}_{10} + \mathbf{e}_1\mathbf{X}_{20})\mathbf{R}_{j1}\| < C\cdot\alpha q\cdot 24\sqrt{6}\cdot\bar{m}\cdot\sqrt{2\bar{m}nk}\cdot\sqrt{s_1(\mathbf{R}_{i1})^2+1}\cdot r^3$, where $C \approx \frac{1}{2\pi}$.

The singular value for matrix $\mathbf{X}_{01}$ which was sampled with parameter $s\sqrt{\bar{m}/2}$ (the same holds for $\mathbf{X}_{11}, \mathbf{X}_{21}, \mathbf{X}_{02}, \mathbf{X}_{12}, \mathbf{X}_{22}$) satisfies $s_1(\mathbf{X}_{01}) \le 2\sqrt{3}\cdot\bar{m}\cdot\sqrt{s_1(\mathbf{R}_{i1})^2+1}\cdot r$. Using the fact $\bar{m} = O(nk), m = \bar{m}+nk$ *and* $s_1(\mathbf{R}_{i1}) \le O(\sqrt{nk})\cdot r$ (from Lemma 4), we finally have

$$\|\tilde{\mathbf{e}}_0\mathbf{R}_{j1} + \tilde{\mathbf{e}}_1\| < \alpha q\cdot O(nk)^3\cdot r^3.$$

Hence, $\tilde{\mathbf{e}}_0\mathbf{R}_{j1}+\tilde{\mathbf{e}}_1,\ \tilde{\mathbf{e}}_0\mathbf{R}_{j2}+\tilde{\mathbf{e}}_2 \in \mathscr{P}_{1/2}(q\cdot\mathbf{B}^{-t})$, by taking $1/\alpha = O(nk)^3\cdot r^3$. □

**Theorem 17** (Security)**.** *The above scheme is* IND-sID-CPA *secure assuming the hardness of decision-*$\mathrm{LWE}_{q,\alpha'}$ *for* $\alpha' = \alpha/3 \ge 2\sqrt{n}/q$.

**Proof:** First, using the same technique in [25], we transform the samples from LWE distribution to what we will need below. Given access to an LWE distribution $\mathbf{A}_{s,\alpha'}$ over $\mathbb{Z}_q^n\times\mathbb{T}$, (where$\mathbb{T} = \mathbb{R}/\mathbb{Z}$) for any $\mathbf{s}\in\mathbb{Z}_q^n$, we can transform its samples $(\mathbf{a}, b = \langle\mathbf{s},\mathbf{a}\rangle/q + e \mod 1)$ to have the form $(\mathbf{a}, 2(\langle\mathbf{s},\mathbf{a}\rangle \mod q) + e' \mod 2q)$ for $e' \leftarrow D_{\mathbb{Z},\alpha q}$, by mapping $b \mapsto 2qb + D_{\mathbb{Z}-2qb,s} \mod 2q$, where $s^2 = (\alpha q)^2 - (2\alpha' q)^2 \ge 4n \ge \eta_\varepsilon(\mathbb{Z})^2$, $\eta_\varepsilon$ is smoothing parameter [27, 25]. This transformation maps the uniform distribution over $\mathbb{Z}_q^n\times\mathbb{T}$ to the uniform distribution $\mathbb{Z}_q^n\times\mathbb{Z}_{2q}$. Once the LWE samples are of the desired form, we construct column-wise matrix $\mathbf{A}^*$ from these samples $\mathbf{a}$ and a vector $\mathbf{b}^*$ from the corresponding b. Let $id_{i^*}$ be the target user. The proof follows by sequence of games.

**Game 0:** This is the original IND-sID-CPA game from definition between an attacker $\mathscr{A}$ against scheme and an IND-sID-CPA challenger.

**Game 1:** In **Game1**, we change the way that the challenger generates $\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2$ in the public parameters. In **SetUp** phase, do as follows:

- Set the public parameter $\bar{\mathbf{A}} = \mathbf{A}^*$, where $\mathbf{A}^*$ is from LWE instance $(\mathbf{A}^*, \mathbf{b}^*)$ and set $\bar{\mathbf{A}}' = -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G}$, where $\mathbf{R}$ is chosen according to **Game 0**.
- Choose four invertible matrices $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4$ uniformly random from $\mathbb{Z}_q^{n\times n}$.
- Choose $\mathbf{R}_{i^*1}, \mathbf{R}_{i^*2} \leftarrow \mathscr{D} = D_{\mathbb{Z},r}^{m\times nk}$; Set $\mathbf{A}'_1 = -\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*1}$ and $\mathbf{A}'_2 = -\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*2}$; Construct $\mathbf{A}_1 = \mathbf{A}'_1 - \mathbf{H}_3\mathbf{H}_{id_{i^*}}\mathbf{G}$ and $\mathbf{A}_2 = \mathbf{A}'_2 - \mathbf{H}_4\mathbf{H}_{id_{i^*}}\mathbf{G}$.
- Set $PP = (\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4, \mathbf{G})$ and send it to $\mathscr{A}$.

To answer secret key query against $id_i \in CU$, challenger will construct $\tilde{\mathbf{A}}_i = \left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G} + \mathbf{H}_{id_i}\mathbf{G}\ \right] = \left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R} + (\mathbf{H}_{id_i} - \mathbf{H}_{id_{i^*}})\mathbf{G}\ \right]$. So, $\mathbf{R}$ is a trapdoor of $\tilde{\mathbf{A}}_i$ with invertible tag $(\mathbf{H}_{id_i} - \mathbf{H}_{id_{i^*}})$. Then using **Extract** algorithm, challenger gets the secret key $sk_{id_i} = \left[\ \mathbf{R}_{i1} \ \middle|\ \mathbf{R}_{i2}\ \right]$ for $id_i$, sends $sk_{id_i}$ to $\mathscr{A}$. Challenger will send $\perp$, against the secret key query for $id_i \in HU$.

Note that for $id_{i^*}$, $\tilde{\mathbf{A}}_{i^*} = \left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]$, so $\mathbf{A}'_1 = -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1}, \mathbf{A}'_2 = -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2}$ and

$\mathbf{A}_{i^*} = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}_{i^*1} \ \middle|\ \mathbf{A}_{i^*2}\ \right] = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}_1 + \mathbf{H}_3\mathbf{H}_{id_{i^*}}\mathbf{G} \ \middle|\ \mathbf{A}_2 + \mathbf{H}_4\mathbf{H}_{id_{i^*}}\mathbf{G}\ \right]$
$= \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}'_1 \ \middle|\ \mathbf{A}'_2\ \right] = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2}\ \right]$.

For the re-encryption key query and re-encryption query, challenger maintain the restrictions as in definition 10 and computes $rk_{i\to j}$, $\mathbf{ReEnc}(rk_{i\to j}, ct)$ according to the **ReKeyGen** and **ReEnc** algorithms to reply the adversary. Due to left-over hash lemma [1, Lemma 14], $(\mathbf{A}^*, -\mathbf{A}^*\mathbf{R}, -\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*1}$, $-\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*2})$ is statistically indistinguishable with uniform distribution. Hence, $(\mathbf{A}^*, -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G}, -\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*1} - \mathbf{H}_3\mathbf{H}_{id_{i^*}}\mathbf{G}$,

$-\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]\cdot\mathbf{R}_{i^*2}-\mathbf{H}_4\mathbf{H}_{id_{i^*}}\mathbf{G}$) is statistically indistinguishable with uniform distribution. Since $\bar{\mathbf{A}},\bar{\mathbf{A}}',\mathbf{A}_1,\mathbf{A}_2$ and responses to key queries are statistically close to those in **Game 0**, **Game 0** and **Game 1** are statistically indistinguishable.

**Game 2:** In **Game2** we change the way that the challenger generates challenge ciphertext. Here Challenger will produce the challenge ciphertext $\mathbf{b}$ on a message $\mathbf{m}\in\{0,1\}^{nk}$ for $id_{i^*}$ as follows: Choose $\mathbf{s}\leftarrow\mathbb{Z}_q^n$ and $\bar{\mathbf{e}}_0\leftarrow D^{\bar{m}}_{\mathbb{Z},\alpha q}$ as usual, but do not choose $\mathbf{e}'_0,\mathbf{e}_1,\mathbf{e}_2$. Let $\bar{\mathbf{b}}_0^t=2(\mathbf{s}^t\mathbf{A}^*\mod q)+\bar{\mathbf{e}}_0^t\mod 2q$ and $\bar{\mathbf{b}'_0}^t=-\bar{\mathbf{b}}_0^t\mathbf{R}+\hat{\mathbf{e}}_0^t\mod 2q$, where $\hat{\mathbf{e}}_0\leftarrow D^{nk}_{\mathbb{Z},s'}$. So, $\mathbf{b}_0=(\bar{\mathbf{b}}_0,\mathbf{b}'_0)$. The last $2nk$ coordinates can be set as $\mathbf{b}_1^t=-\mathbf{b}_0^t\mathbf{R}_{i^*1}+\hat{\mathbf{e}}_1^t\mod 2q$; $\mathbf{b}_2^t=-\mathbf{b}_0^t\mathbf{R}_{i^*2}+\hat{\mathbf{e}}_2^t+encode(\mathbf{m})\mod 2q$, where $\hat{\mathbf{e}}_1,\hat{\mathbf{e}}_2\leftarrow D^{nk}_{\mathbb{Z},s'}$. Finally, replace $\bar{\mathbf{b}}_0$ with $\mathbf{b}^*$ in all the above expression, where $(\mathbf{A}^*,\mathbf{b}^*)$ is the LWE instance. Therefore, $\bar{\mathbf{b}_0^*}^t=\mathbf{b}^{*t}$; $\bar{\mathbf{b}'_0}^t=-\mathbf{b}^{*t}\mathbf{R}+\hat{\mathbf{e}}_0^t\mod 2q$; $\mathbf{b}_1^t=-\mathbf{b}_0^{*t}\mathbf{R}_{i^*1}+\hat{\mathbf{e}}_1^t\mod 2q$; $\mathbf{b}_2^t=-\mathbf{b}_0^{*t}\mathbf{R}_{i^*2}+\hat{\mathbf{e}}_2^t+encode(\mathbf{m})\mod 2q$. Set $\mathbf{b}_0^{*t}=(\mathbf{b}^{*t},-\mathbf{b}^{*t}\mathbf{R}+\hat{\mathbf{e}}_0^t\mod 2q)$. Then the challenger output the challenge ciphertext $ct=\mathbf{b}=(\mathbf{b}_0^*,\mathbf{b}_1,\mathbf{b}_2)$.

We now show that the distribution of $\mathbf{b}$ is within $\mathsf{negl}(n)$ statistical distance of that in **Game 1** from the adversary's view. Clearly, $\mathbf{b}^*$ have essentially the same distribution as in **Game 0** by construction. By substitution we have: $\bar{\mathbf{b}'_0}^t=2(\mathbf{s}^t(-\mathbf{A}^*\mathbf{R})\mod q)+\bar{\mathbf{e}}_0^t\mathbf{R}+\hat{\mathbf{e}}_0^t\mod 2q$; $\mathbf{b}_1^t=2(\mathbf{s}^t(-\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1})\mod q)+(\bar{\mathbf{e}}_0^t,\bar{\mathbf{e}}_0^t\mathbf{R}+\hat{\mathbf{e}}_0^t)\mathbf{R}_{i^*1}+\hat{\mathbf{e}}_1^t\mod 2q$; $\mathbf{b}_2^t=2(\mathbf{s}^t(-\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2})\mod q)+(\bar{\mathbf{e}}_0^t,\bar{\mathbf{e}}_0^t\mathbf{R}+\hat{\mathbf{e}}_0^t)\mathbf{R}_{i^*2}+\hat{\mathbf{e}}_2^t+encode(\mathbf{m})\mod 2q$.

By Corollary 3.10 in [30], the noise term $\bar{\mathbf{e}}_0^t\mathbf{R}+\hat{\mathbf{e}}_0^t$ of $\mathbf{b}'_0$ is within $\mathsf{negl}(n)$ statistical distance from discrete Gaussian distribution $D^{nk}_{\mathbb{Z},s'}$. The same argument, also, applies for the noise term of $\mathbf{b}_1,\mathbf{b}_2$. Hence, **Game 1** and **Game 2** are statistically indistinguishable.

**Game 3:** Here, we only change how the $\mathbf{b}^*$ component of the challenge ciphertext is created, letting it be uniformly random in $\mathbb{Z}_{2q}^{\bar{m}}$. Challenger construct the public parameters, answer the secret key queries, re-encryption queries and construct the last $3nk$ coordinates of challenge ciphertext exactly as in Game 2. It follows from the hardness of the decisional $\mathrm{LWE}_{q,\alpha'}$ that **Game 2** and **Game 3** are computationally indistinguishable.

Now by the left-over hash lemma [1, Lemma 14], $(\mathbf{A}^*,\mathbf{b}^*,-\mathbf{A}^*\mathbf{R},\mathbf{b}^{*t}\mathbf{R},-\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1},\mathbf{b}_0^{*t}\mathbf{R}_{i^*1},-\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2},\mathbf{b}_0^{*t}\mathbf{R}_{i^*2})$ is $\mathsf{negl}(n)$-uniform when $\mathbf{R},\mathbf{R}_{i^*1},\mathbf{R}_{i^*2}$ are chosen as in Game 2. Therefore, the challenge ciphertext has the same distribution (up to $\mathsf{negl}(n)$ statistical distance) for any encrypted message. So, the advantage of the adversary against the proposed scheme is same as the advantage of the attacker against decisional $\mathrm{LWE}_{q,\alpha'}$. □

# 4 Single-hop Selective Identity-Based Unidirectional Proxy Re-Signature Scheme (IB-uPRS)

## 4.1 Construction of Single-hop IB-uPRS

In this section, we present our construction of single-hop IB-uPRS. We set the parameters as the following.

- $\mathbf{G}\in\mathbb{Z}_q^{n\times nk}$ is a gadget matrix for large enough prime power $q=poly(n)$ and $k=O(\log q)=O(\log n)$, so there are efficient algorithms to invert $g_{\mathbf{G}}$ and to sample for $f_{\mathbf{G}}$.
- $\bar{m}=O(nk)$ and the Gaussian $\mathscr{D}=D^{\bar{m}\times nk}_{\mathbb{Z},r}$, so that $(\bar{\mathbf{A}},\bar{\mathbf{A}}\mathbf{R})$ is negl(n)-far from uniform for $\bar{\mathbf{A}}\leftarrow\mathbb{Z}_q^{n\times\bar{m}}$ and $\mathbf{R}\leftarrow\mathscr{D}$.
- Let the message space $\mathscr{M}$ be $\{0,1\}^l$, where $l=O((nk)^6)$.
- The real number $\beta_{SIS}$ of $\mathrm{SIS}_{q,n,\bar{m},\beta_{SIS}}$ for IB-uPRS should satisfy $q\geq\beta_{SIS}\omega(\sqrt{n\log n})$ [27], and we set $\beta_{SIS}=O((nk)^{3.5})\cdot r^6$.
- **Encoding of Identity:** We use the FRD map same as in IB-uPRE, to map the identity $id$ to an invertible matrix $\mathbf{H}_{id}$.

- **Encoding of Sign-tag:** We consider the Sign-tag space $\mathscr{T} = \mathbb{Z}_q^n$. We map a non-zero Sign-tag $\mathbf{t} \in \mathscr{T}$ to an invertible matrix $\mathbf{H_t}$, using the same FRD map.

Our construction of IB-uPRS is as follows:

**SetUp**$(1^n)$ :

1. Choose $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n\times\bar{m}}$, $\mathbf{R} \leftarrow \mathscr{D}$, and construct $\bar{\mathbf{A}}' = -\bar{\mathbf{A}}\mathbf{R} \in \mathbb{Z}_q^{n\times nk}$.
2. Choose an invertible matrix $\mathbf{H}_1$ at uniformly random from $\mathbb{Z}_q^{n\times n}$.
3. Choose two random matrices $\mathbf{A}_1, \mathbf{A}_2$ from $\mathbb{Z}_q^{n\times nk}$.
4. Choose uniformly random $(l+1)$ vectors $\mathbf{a}$ and $\mathbf{b}_1, \cdots, \mathbf{b}_l$ from $\mathbb{Z}_q^n$.
5. Output public parameters $PP = (\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2, \mathbf{H}_1, \mathbf{a}, \mathbf{b}_1, \cdots, \mathbf{b}_l, \mathbf{G})$ and the master secret key $msk = \mathbf{R}$.

**Extract**$(PP, msk, id)$ : Let $id_i$ be the identity of $i$-th user.

1. Construct $\tilde{\mathbf{A}}_i = \left[\begin{array}{c|c|c} \bar{\mathbf{A}} & \bar{\mathbf{A}}' + \mathbf{H}_{id_i}\mathbf{G} \end{array}\right] = \left[\begin{array}{c|c|c} \bar{\mathbf{A}} & -\bar{\mathbf{A}}\mathbf{R} + \mathbf{H}_{id_i}\mathbf{G} \end{array}\right] \in \mathbb{Z}_q^{n\times m}$, where $m = \bar{m} + nk$. So, $\mathbf{R}$ is a trapdoor of $\tilde{\mathbf{A}}_i$ with tag $\mathbf{H}_{id_i}$.
2. - Construct $\mathbf{A}_{i1} = \mathbf{A}_1 + \mathbf{H}_{id_i}\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ and set $\mathbf{A}'_{i1} = \left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} \end{array}\right] \in \mathbb{Z}_q^{n\times(m+nk)}$.
   - Call the algorithm $\mathbf{DelTrap}^{\mathscr{O}}(\mathbf{A}'_{i1}, \mathbf{R}, \mathbf{H}_1, s)$ to get a trapdoor $\mathbf{R}_{i1} \in \mathbb{Z}^{m\times nk}$ for $\mathbf{A}'_{i1}$ with tag $\mathbf{H}_1 \in \mathbb{Z}_q^{n\times n}$, where $s \geq \eta_\varepsilon(\Lambda^\perp(\tilde{\mathbf{A}}_i))$; i.e., using $\mathbf{Sample}^{\mathscr{O}}$, sample each column of $\mathbf{R}_{i1}$ independently from a discrete Gaussian with parameter $s$ over the appropriate coset of $\Lambda^\perp(\tilde{\mathbf{A}}_i)$, so that $\tilde{\mathbf{A}}_i\mathbf{R}_{i1} = \mathbf{H}_1\mathbf{G} - \mathbf{A}_{i1}$.
3. - Construct $\mathbf{A}_{i2} = \mathbf{A}_2 + \mathbf{H}_{id_i}\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ and set $\mathbf{A}'_{i2} = \left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i2} \end{array}\right] \in \mathbb{Z}_q^{n\times(m+nk)}$.
   - Call the algorithm $\mathbf{DelTrap}^{\mathscr{O}}$ to get a trapdoor $\mathbf{R}_{i2} \in \mathbb{Z}^{m\times nk}$ for $\mathbf{A}'_{i2}$ with tag $\mathbf{0} \in \mathbb{Z}_q^{n\times n}$; i.e., using $\mathbf{Sample}^{\mathscr{O}}$, sample each column of $\mathbf{R}_{i2}$ independently from a discrete Gaussian with parameter $s$ over the appropriate coset of $\Lambda^\perp(\tilde{\mathbf{A}}_i)$, so that $\tilde{\mathbf{A}}_i\mathbf{R}_{i2} = -\mathbf{A}_{i2}$.

   Output the signing key as $sk_{id_i} = \left[\begin{array}{c|c} \mathbf{R}_{i1} & \mathbf{R}_{i2} \end{array}\right] \in \mathbb{Z}^{m\times 2nk}$.

**Sign**$(PP, sk_{id_i}, \mathbf{m} \in \{0,1\}^l)$ :

1. Randomly choose a non-zero Sign-tag $\mathbf{t} \in \mathscr{T}$, construct $\mathbf{H_t}$.
2. Construct $\tilde{\mathbf{A}}_i$, $\mathbf{A}_{i1}$, and $\mathbf{A}_{i2}$ for $id_i$ same as in **Extract** algorithm.
3. Set the signing matrix:
   $\mathbf{A}_{id_i,\mathbf{t}} = \left[\begin{array}{c|c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i1} & \mathbf{A}_{i2} + \mathbf{H_t} \end{array}\right] = \left[\begin{array}{c|c|c} \tilde{\mathbf{A}}_i & -\tilde{\mathbf{A}}_i\mathbf{R}_{i1} + \mathbf{H}_1\mathbf{G} & -\tilde{\mathbf{A}}_i\mathbf{R}_{i2} + \mathbf{H_t} \end{array}\right]$.
4. Using the $l$-bit message $\mathbf{m} = (m_1, \cdots, m_l)$, where $m_i \in \{0,1\}$; construct $\mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i \in \mathbb{Z}_q^n$.
5. Sample the vector $(\mathbf{e}_0, \mathbf{e}_2) \in \mathbb{Z}^m \times \mathbb{Z}^{nk}$ for the cosets obtained from $\mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i - (-\tilde{\mathbf{A}}_i\mathbf{R}_{i1} + \mathbf{H}_1\mathbf{G})\mathbf{e}_1$; we use $\mathbf{Sample}^{\mathscr{O}}$ for the matrix $\left[\begin{array}{c|c} \tilde{\mathbf{A}}_i & \mathbf{A}_{i2} + \mathbf{H_t} \end{array}\right]$ with the generalized trapdoor $\mathbf{R}_{i2}$ and tag $\mathbf{H_t}$ with parameter $s = O(\sqrt{nk}) \cdot r^2$. It holds that $\mathbf{A}_{id_i,\mathbf{t}} \cdot \mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$, where $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{Z}^{m+2nk}$.
6. Output the signature $\mathbf{e}$ together with the corresponding Sign-tag $\mathbf{t}$.

**Verify**$(PP, id_i, \mathbf{m}, \mathbf{e}, \mathbf{t})$ :

1. Reconstruct the signing matrix
   $\mathbf{A}_{id_i,\mathbf{t}} = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i1} \ \middle|\ \mathbf{A}_{i2}+\mathbf{H}_t\ \right] = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ -\tilde{\mathbf{A}}_i\mathbf{R}_{i1}+\mathbf{H}_1\mathbf{G} \ \middle|\ -\tilde{\mathbf{A}}_i\mathbf{R}_{i2}+\mathbf{H}_\mathbf{t}\ \right]$.
2. Accept if $\|\mathbf{e}\| \le \beta_{max}$ and $\mathbf{A}_{id_i,\mathbf{t}} \cdot \mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$, where $\beta_{max} = \bar{m}\sqrt{m+2nk} \cdot s^2$; otherwise, reject.

**ReKeyGen**$(PP, id_i, id_j, sk_{id_j})$ : Let $sk_{id_j} = \left[\ \mathbf{R}_{j1} \ \middle|\ \mathbf{R}_{j2}\ \right]$ be the signing key for $j$-th user.

1. Construct $\mathbf{A}_{id_i} = \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_{i1} \ \middle|\ \mathbf{A}_{i2}\ \right]$, where $\tilde{\mathbf{A}}_i = \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G}\ \right]$ and $\mathbf{A}_{i1}, \mathbf{A}_{i2}$ are same as in **Extract** algorithm.
2. Construct $\mathbf{A}_{id_j} = \left[\ \tilde{\mathbf{A}}_j \ \middle|\ \mathbf{A}_{j1} \ \middle|\ \mathbf{A}_{j2}\ \right]$, where $\tilde{\mathbf{A}}_j = \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G}\ \right]$ and $\mathbf{A}_{j1}, \mathbf{A}_{j2}$ are same as in **Extract** algorithm.
3. Using **Sample**$^{\mathcal{O}}$ with trapdoor $\mathbf{R}_{j1}$(from the signing key of $j$th user ), with tag $\mathbf{H}_1$, we sample from the cosets which are formed with the column of the matrix $\bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G}$. After sampling $nk$ times we get an $(\bar{m}+2nk) \times nk$ matrix and parse it as three matrices $\mathbf{X}_{00} \in \mathbb{Z}^{\bar{m}\times nk}$, $\mathbf{X}_{10} \in \mathbb{Z}^{nk\times nk}$ and $\mathbf{X}_{20}$ $\in \mathbb{Z}^{nk\times nk}$ matrices with Gaussian entries of parameter $s$. So, $\left[\ \tilde{\mathbf{A}}_j \ \middle|\ -\tilde{\mathbf{A}}_j\mathbf{R}_{j1}+\mathbf{H}_1\mathbf{G}\ \right] \begin{bmatrix} \mathbf{X}_{00} \\ \mathbf{X}_{10} \\ \mathbf{X}_{20} \end{bmatrix} =$ $\bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G}$. *i.e.* $\left[\ \tilde{\mathbf{A}}_j \ \middle|\ \mathbf{A}_{j1}\ \right] \begin{bmatrix} \mathbf{X}_{00} \\ \mathbf{X}_{10} \\ \mathbf{X}_{20} \end{bmatrix} = \bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G}$.
4. Continue sampling for the cosets obtained from the columns of the matrix $\mathbf{A}_{i1}$. This time, we increase the Gaussian parameter of the resulting sampled matrix up to $s\sqrt{\bar{m}/2}$:
   $$\left[\ \tilde{\mathbf{A}}_j \ \middle|\ -\tilde{\mathbf{A}}_j\mathbf{R}_{j1}+\mathbf{H}_1\mathbf{G}\ \right] \begin{bmatrix} \mathbf{X}_{01} \\ \mathbf{X}_{11} \\ \mathbf{X}_{21} \end{bmatrix} = \mathbf{A}_{i1} \text{ , } \textit{i.e.} \left[\ \tilde{\mathbf{A}}_j \ \middle|\ \mathbf{A}_{j1}\ \right] \begin{bmatrix} \mathbf{X}_{01} \\ \mathbf{X}_{11} \\ \mathbf{X}_{21} \end{bmatrix} = \mathbf{A}_{i1} \ .$$
   For the last sampling, to get a correct re-signing key, we use the cosets which are formed with the column of the matrix $\mathbf{A}_{i2}-\mathbf{A}_{j2}$: $\left[\ \tilde{\mathbf{A}}_j \ \middle|\ \mathbf{A}_{j1}\ \right] \begin{bmatrix} \mathbf{X}_{02} \\ \mathbf{X}_{12} \\ \mathbf{X}_{22} \end{bmatrix} = \mathbf{A}_{i2}-\mathbf{A}_{j2}$, where $\mathbf{X}_{01}, \mathbf{X}_{02} \in \mathbb{Z}^{\bar{m}\times nk}$, $\mathbf{X}_{11}, \mathbf{X}_{12}, \mathbf{X}_{21}, \mathbf{X}_{22} \in \mathbb{Z}^{nk\times nk}$ with entries distributed as Gaussian with parameter $s\sqrt{\bar{m}}$.
5. The re-signing key is a matrix with Gaussian entries:
   $$rk_{i\to j} = \begin{bmatrix} \mathbf{I}_{\bar{m}\times\bar{m}} & \mathbf{X}_{00} & \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{0} & \mathbf{X}_{10} & \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{0} & \mathbf{X}_{20} & \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{nk\times nk} \end{bmatrix} \in \mathbb{Z}^{(m+2nk)\times(m+2nk)} \text{ and it satisfies:}$$
   $\left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}'+\mathbf{H}_{id_j}\mathbf{G} \ \middle|\ \mathbf{A}_{j1} \ \middle|\ \mathbf{A}_{j2}\ \right] \cdot rk_{i\to j} = \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}'+\mathbf{H}_{id_i}\mathbf{G} \ \middle|\ \mathbf{A}_{i1} \ \middle|\ \mathbf{A}_{i2}\ \right]$.
   Notice that, it satisfies $\mathbf{A}_{id_j,\mathbf{t}} \cdot rk_{i\to j} = \mathbf{A}_{id_i,\mathbf{t}}$ for any Sign-tag $\mathbf{t}$.

**ReSign**$(PP, rk_{i\to j}, id_i, id_j, \mathbf{e}, \mathbf{m}, \mathbf{t})$ :

1. Ouput $\perp$, if **Verify**$(PP, id_i, \mathbf{m}, \mathbf{e}, \mathbf{t}) \to$ reject.
2. Otherwise, compute $\mathbf{e}' = rk_{i\to j} \cdot \mathbf{e} \in \mathbb{Z}^{m+2nk}$. Output re-signature $\mathbf{e}'$ for the message $\mathbf{m}$ with the same Sign-tag $\mathbf{t}$ under $id_j$.

### 4.2 Correctness and Security

**Theorem 18** (Correctness)**.** *The* IB-uPRS *scheme with parameters proposed in Section 4.1 is correct.*

**Proof:** Let $sk_{id_i} = \left[\begin{array}{c|c} \mathbf{R}_{i1} & \mathbf{R}_{i2} \end{array}\right]$ and $sk_{id_j} = \left[\begin{array}{c|c} \mathbf{R}_{j1} & \mathbf{R}_{j2} \end{array}\right]$ are the signing key for $i$, $j$-th user respectively. Let the Re-signing key from $i$th to $j$th user be

$$\left[\begin{array}{cccc} \mathbf{I}_{\bar{m}\times\bar{m}} & \mathbf{X}_{00} & \mathbf{X}_{01} & \mathbf{X}_{02} \\ \mathbf{0} & \mathbf{X}_{10} & \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{0} & \mathbf{X}_{20} & \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{nk\times nk} \end{array}\right].$$

As $\|\mathbf{e}\| \leq s\cdot\sqrt{m+2nk}$, it follows from the construction that if a signer construct a signature $\mathbf{e}$ on a message $\mathbf{m}$ with Sign-tag $\mathbf{t}$, under the identity $id_i$, it will be accepted by the verifier.

Let $\mathbf{e}'$ be the re-signature of $\mathbf{m}$ with the same Sign-tag $\mathbf{t}$ under the $j$-th user. Now for the re-signature, we need to snow that $\mathbf{A}_{id_j,\mathbf{t}}\cdot\mathbf{e}' = \mathbf{a} + \sum_{i=1}^{l} m_i\cdot\mathbf{b}_i$, *and* $\|\mathbf{e}'\| \leq \beta_{max}$.

We have,

$$\begin{aligned}\mathbf{A}_{id_j,\mathbf{t}}\cdot\mathbf{e}' &= \mathbf{A}_{id_j,\mathbf{t}}\cdot(rk_{i\to j}\cdot\mathbf{e}) = (\mathbf{A}_{id_j,\mathbf{t}}\cdot rk_{i\to j})\cdot\mathbf{e} \\ &= \mathbf{A}_{id_i,\mathbf{t}}\cdot\mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i\cdot\mathbf{b}_i, \text{ as } \mathbf{Verify}(PP, id_i, \mathbf{m}, \mathbf{e}, \mathbf{t}) \to accept.\end{aligned}$$

Now $\|\mathbf{e}'\| = \|rk_{i\to j}\cdot\mathbf{e}\| \leq \|rk_{i\to j}\|\cdot\|\mathbf{e}\| \leq s_1(rk_{i\to j})\cdot\|\mathbf{e}\|$. We have $\|\mathbf{e}'\| \leq \bar{m}\sqrt{m+2nk}\cdot s^2 = \beta_{max}$. This completes the proof. □

**Theorem 19** (Security)**.** *The above scheme is existential unforgeable against adaptive chosen message and selective identity attacks in the standard model under the hardness of* $\mathrm{SIS}_{q,n,\bar{m},\beta_{SIS}}$.

**Proof:** Let a reduction $\mathscr{C}$ attacking SIS. Let $\mathbf{A}^* \in \mathbb{Z}_q^{n\times\bar{m}}$ be the SIS instance, to which, asked to return a solution $\mathbf{e} \in \mathbb{Z}_q^{\bar{m}}$, such that $\mathbf{A}\cdot\mathbf{e} = 0 \mod q$ and $0 \neq \|\mathbf{e}\| \leq \beta_{SIS}$. Let $id^* \in HU$ be the challenge identity and $\mathbf{t}^* \in \mathscr{T}$ be the challenge Sign-tag, send by the adversary at the begining of the game. The proof proceeds in a sequence of games.

**Game 0:** This is the original security game from definition between an adversary $\mathscr{A}$ and $\mathscr{C}$.

**Game 1:**We set $\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{a}, \mathbf{b}_1, \cdots, \mathbf{b}_l$ from public parameter as follows:
$\bar{\mathbf{A}} = \mathbf{A}^*$, where $\mathbf{A}^*$ is from SIS instance and set $\bar{\mathbf{A}}' = -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G}$, where $\mathbf{R}$ is chosen in the same way as in **Game 0**.

Now choose $(l+1)$ small vectors $\mathbf{s}_0, \mathbf{s}_1, \cdots, \mathbf{s}_l \in \mathscr{D}_{\mathbb{Z}^m,s}$, set $\mathbf{a} = \mathbf{A}^*\mathbf{s}_0$ and $\mathbf{b}_i = \mathbf{A}^*\mathbf{s}_i$. Due to left-over hash lemma [1, Lemma 14], $(\mathbf{A}^*, -\mathbf{A}^*\mathbf{R}, \mathbf{A}^*\mathbf{s}_0, \mathbf{A}^*\mathbf{s}_i)$ is statistically indistinguishable with uniform distribution. Hence, $(\mathbf{A}^*, -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G}, \mathbf{A}^*\mathbf{s}_0, \mathbf{A}^*\mathbf{s}_i)$ is statistically indistinguishable with uniform distribution. So, **Game 0** and **Game 1** are statistically indistinguishable to adversary $\mathscr{A}$.

Choose an invertible matrix $\mathbf{H}_1$ at uniformly random from $\mathbb{Z}_q^{n\times n}$. Set $\mathbf{A}' = \left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} \end{array}\right]$ and choose $\mathbf{R}_{i^*1}, \mathbf{R}_{i^*2} \leftarrow \mathscr{D} = D_{\mathbb{Z},r}^{m\times nk}$. Set $\mathbf{A}'_1 = -\mathbf{A}'\cdot\mathbf{R}_{i^*1}$ and $\mathbf{A}'_2 = -\mathbf{A}'\cdot\mathbf{R}_{i^*2}$. Construct $\mathbf{A}_1 = \mathbf{A}'_1 - \mathbf{H}_{id_{i^*}}\mathbf{G}$ and $\mathbf{A}_2 = \mathbf{A}'_2 - \mathbf{H}_{id_{i^*}}\mathbf{G} - \mathbf{H}_{t^*}\mathbf{G}$. Set $PP = (\bar{\mathbf{A}}, \bar{\mathbf{A}}', \mathbf{A}_1, \mathbf{A}_2, \mathbf{H}_1, \mathbf{a}, \mathbf{b}_1, \cdots \mathbf{b}_l, \mathbf{G})$ and send it to the Adversary $\mathscr{A}$.

$\mathscr{O}^{\mathbf{Extract}}$: To answer a signing key query against $id_i (\neq id_{i^*})$, challenger will do as follows: Construct $\tilde{\mathbf{A}}_i = \left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} - \mathbf{H}_{id_{i^*}}\mathbf{G} + \mathbf{H}_{id_i}\mathbf{G} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}^* & -\mathbf{A}^*\mathbf{R} + (\mathbf{H}_{id_i} - \mathbf{H}_{id_{i^*}})\mathbf{G} \end{array}\right]$. So, $\mathbf{R}$ is a trapdoor of $\tilde{\mathbf{A}}_i$ with invertible tag $(\mathbf{H}_{id_i} - \mathbf{H}_{id_{i^*}})$.

Then using **Extract** algorithm, challenger gets the signing key $sk_{id_i} = \left[\begin{array}{c|c} \mathbf{R}_{i1} & \mathbf{R}_{i2} \end{array}\right]$ for $id_i$, sends $sk_{id_i}$ to the adversary $\mathscr{A}$.

$\mathscr{O}^{\mathbf{Sign}}$: For sign-query of message $\mathbf{m}$ under $id_i (\neq id_{i^*})$, first selects a non-zero Sign-tag $\mathbf{t}$ such that $\mathbf{t} \neq \mathbf{t}^*$, then compute the signature $\mathbf{e}$ of $\mathbf{m}$ with Sign-tag $\mathbf{t}$ according to the **Sign** algorithm and sends it to

the adversary. Sign-query of message $\mathbf{m}$ under $id_{i^*}$, with Sign-tag $\mathbf{t}$ such that $\mathbf{t} \neq \mathbf{t}^*$, the signing matrix $\mathbf{A}_{id_{i^*},\mathbf{t}} = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}_{i^*1} \ \middle|\ \mathbf{A}_{i^*2} + \mathbf{H}_{\mathbf{t}}\ \right]$
$= \left[\ \tilde{\mathbf{A}}_i \ \middle|\ \mathbf{A}_1 + \mathbf{H}_{id_{i^*}}\mathbf{G} \ \middle|\ \mathbf{A}_2 + \mathbf{H}_{id_{i^*}}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right] = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}'_1 \ \middle|\ \mathbf{A}'_2 - \mathbf{H}_{\mathbf{t}^*}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right]$
$= \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2} - \mathbf{H}_{t^*}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right]$.
So, for $\left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2} - \mathbf{H}_{\mathbf{t}^*}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right]$, $\mathbf{R}_{i^*2}$ is a trapdoor with invertible tag $\mathbf{H}_{\mathbf{t}} - \mathbf{H}_{\mathbf{t}^*}$. So using this we can sign a message under $id_{i^*}$ with Sign-tag $\mathbf{t} \neq \mathbf{t}^*$.

$\mathscr{O}^{\mathbf{ReKeyGen}}$: For the re-signing key query, challenger maintain the restrictions as in definition 14 and computes $rk_{i\to j}$, according to the **ReKeyGen** algorithm to reply the adversary.

$\mathscr{O}^{\mathbf{ReSign}}$: For a re-signature query on the signarure $\mathbf{e}$ of the message $\mathbf{m} = (m_1, \cdots, m_l)$, with Sign-tag $\mathbf{t}$ under $id_i$, to a signature under $id_j$, we consider following cases:

- If $\mathbf{t} = \mathbf{t}^*$, or $\mathbf{Verify}(PP, id_i, \mathbf{m}, \mathbf{e}, \mathbf{t})$= reject, output $\perp$.
- If $id_i \in HU$, $id_j \in HU \setminus \{id^*\}$ or $id_i, id_j \in CU$ or $id_i \in HU, id_j \in CU$, output the re-signature using **ReSign** algorithm.
- For re-signing the signarure $\mathbf{e}$ of the message $\mathbf{m}$, with Sign-tag $\mathbf{t} \neq \mathbf{t}^*$ under $id_i \in HU$ to $id^*$, do as follows:
  - Construct $\mathbf{A}_{id_{i^*},\mathbf{t}} = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}_{i^*1} \ \middle|\ \mathbf{A}_{i^*2} + \mathbf{H}_{\mathbf{t}}\ \right]$
    $= \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2} - \mathbf{H}_{\mathbf{t}^*}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right]$.
    So, for $\left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2} - \mathbf{H}_{\mathbf{t}^*}\mathbf{G} + \mathbf{H}_{\mathbf{t}}\mathbf{G}\ \right]$, $\mathbf{R}_{i^*2}$ is a trapdoor with invertible tag $\mathbf{H}_{\mathbf{t}} - \mathbf{H}_{\mathbf{t}^*}$. So $\mathbf{R}_{i^*2}$ is a trapdoor for $\mathbf{A}_{id_{i^*},\mathbf{t}}$ (extension), using this we can sample a (pseudo) Re-signing key $rk$ s.t. $\mathbf{A}_{id_{i^*},\mathbf{t}} \cdot rk = \mathbf{A}_{id_i,\mathbf{t}}$.
  - Since, $\mathbf{A}_{id_i,\mathbf{t}} \cdot \mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$, so we have $\mathbf{A}_{id_{i^*},\mathbf{t}} \cdot rk \cdot \mathbf{e} = \mathbf{a} + \sum_{i=1}^{l} m_i \cdot \mathbf{b}_i$.
  - Output the re-signature $\mathbf{e}' = rk \cdot \mathbf{e}$ under $id^*$.

Now adversary $\mathscr{A}$ sends the forgery tuple $(id^*, \mathbf{m}^*, \mathbf{e}^*, \mathbf{t}^*)$ to $\mathscr{C}$, where $\mathbf{m}^* = (m_1^*, \cdots, m_l^*)$. Note that for $id_{i^*}$, $\tilde{\mathbf{A}}_{i^*} = \left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right]$, which is $\mathbf{A}'$ from the set up phase and $\mathbf{A}_{id_{i^*},\mathbf{t}^*} = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ \mathbf{A}_{i^*1} \ \middle|\ \mathbf{A}_{i^*2} + \mathbf{H}_{\mathbf{t}^*}\ \right] = \left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2}\ \right]$. Now $\mathscr{C}$ constructs a solution to SIS instance $\mathbf{A}^*$ as follows: Parsed $\mathbf{e}^*$ as $\mathbf{e}^* = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{Z}^m \times \mathbb{Z}^{nk} \times \mathbb{Z}^{nk}$. Thus, for correctness, it holds that $\mathbf{A}_{id_{i^*},\mathbf{t}} \cdot \mathbf{e}^* = \mathbf{a} + \sum_{i=1}^{l} m_i^* \cdot \mathbf{b}_i$ i.e., $\left[\ \tilde{\mathbf{A}}_{i^*} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*1} \ \middle|\ -\tilde{\mathbf{A}}_{i^*}\mathbf{R}_{i^*2}\ \right] \cdot (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) = \mathbf{A}^*\mathbf{s}_0 + \sum_{i=1}^{l} m_i^* \cdot \mathbf{A}^*\mathbf{s}_i$ i.e. $\Rightarrow \tilde{\mathbf{A}}_{i^*}(\mathbf{e}_1 - \mathbf{R}_{i^*1}\mathbf{e}_2 - \mathbf{R}_{i^*2}\mathbf{e}_3) = \mathbf{A}^*(\mathbf{s}_0 + \sum_{i=1}^{l} m_i^*\mathbf{s}_i)$, where $(\mathbf{e}_1 - \mathbf{R}_{i^*1}\mathbf{e}_2 - \mathbf{R}_{i^*2}\mathbf{e}_3) \in \mathbb{Z}^m$. Now parsed $(\mathbf{e}_1 - \mathbf{R}_{i^*1}\mathbf{e}_2 - \mathbf{R}_{i^*2}\mathbf{e}_3)$ as $(\mathbf{e}_1^*, \mathbf{e}_2^*) \in \mathbb{Z}^{\bar{m}} \times \mathbb{Z}^{nk}$ and let $\mathbf{S}^* = \mathbf{s}_0 + \sum_{i=1}^{l} m_i^*\mathbf{s}_i$. So, we get $\tilde{\mathbf{A}}_{i^*}(\mathbf{e}_1^*, \mathbf{e}_2^*) = \mathbf{A}^*\mathbf{S}^*$ i.e., $\left[\ \mathbf{A}^* \ \middle|\ -\mathbf{A}^*\mathbf{R}\ \right](\mathbf{e}_1^*, \mathbf{e}_2^*) = \mathbf{A}^*\mathbf{S}^*$ i.e., $\mathbf{A}^*(\mathbf{e}_1^* - \mathbf{R}\mathbf{e}_2^* - \mathbf{S}^*) = 0$.

Since, matrices $\mathbf{R}, \mathbf{R}_{i^*1}, \mathbf{R}_{i^*2}$ and vectors $\mathbf{s}_0, \mathbf{s}_i$ are independent of each other, and hidden from adversary $\mathscr{A}$'s view, then with overwhelming probability $(\mathbf{e}_1^* - \mathbf{R}\mathbf{e}_2^* - \mathbf{S}^*) \neq 0$. As $\|\mathbf{e}^*\| \leq \bar{m}\sqrt{m + 2nk} \cdot s^2 = \beta_{max}$ and $s_1(\mathbf{R}_{i^*1}), s_1(\mathbf{R}_{i^*2}) \leq O(\sqrt{nk}) \cdot r$, we have $\|(\mathbf{e}_1^*, \mathbf{e}_2^*)\| \leq O((nk)^3) \cdot r^5$. Since, $\bar{m} = O(nk)$, $m = \bar{m} + nk$, and $\mathbf{s}_j \leq O(\sqrt{m}) \cdot r$ for $j = 0, 1, \cdots, l$, we have $\|\mathbf{S}^*\| \leq \sqrt{l+1} \cdot O(\sqrt{nk}) \cdot r$. Finally, $s_1(\mathbf{R}) \leq O(\sqrt{nk}) \cdot r$ implies $\|(\mathbf{e}_1^* - \mathbf{R}\mathbf{e}_2^* - \mathbf{S}^*)\| \leq O((nk)^{3.5}) \cdot r^6 + \sqrt{l+1} \cdot O(\sqrt{nk}) \cdot r$. According to the security parameters of section 4.1, $(\mathbf{e}_1^* - \mathbf{R}\mathbf{e}_2^* - \mathbf{S}^*) \leq O((nk)^{3.5}) \cdot r^6$. Hence, $(\mathbf{e}_1^* - \mathbf{R}\mathbf{e}_2^* - \mathbf{S}^*)$ works as the solution for $\mathrm{SIS}_{q,n,\bar{m},\beta_{SIS}}$ instance $\mathbf{A}^*$, where $\beta_{SIS} = O((nk)^{3.5}) \cdot r^6$. This completes the proof. □

# 5 Conclusion

In this paper, we first propose quantum-safe concrete constructions of IB-uPRE and IB-uPRS secure in the standard model. Both IB-uPRE and IB-uPRS enjoy the important features like Non-transitivity, Proxy transparency, Key optimality, and Non-interactivity. The proposed constructions are single-hop. It is an interesting open issue to construct multi-hop version of the proposed schemes.

# References


[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10), Monaco / French Riviera*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer-Verlag, May-June 2010.

[2] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96), Philadelphia, Pennsylvania, USA*, pages 99–108. ACM, July 1996.

[3] M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of the 26th International Colloquium on Automata, Languages, and Programming (ICALP'99), Prague, Czech Republic*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer-Verlag, July 1999.

[4] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, February 2006.

[5] G. Ateniese and S. Hohenberger. Proxy re-signatures: new definitions, algorithms, and applications. In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS'05), Alexandria, Virginia, USA*, pages 310–319. ACM, November 2005.

[6] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, December 1993.

[7] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *Proc. of the 17th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'98), Espoo, Finland*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer-Verlag, May-June 1998.

[8] F. Böhl, D. Hofheinz, T. Jager, J. Koch, J. H. Seo, and C. Striecks. Practical signatures from standard assumptions. In *Proc. of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'13), Athens, Greece*, pages 461–485. Springer-Verlag, May 2013.

[9] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proc. of the 14th ACM Conference on Computer and Communications Security (CCS'07), Alexandria, Virginia, USA*, pages 185–194. ACM, October 2007.

[10] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *Proc. of the 12th Theory of Cryptography Conference (TCC'15), Warsaw, Poland*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer-Verlag, March 2015.

[11] N. Chandran, M. Chase, F. Liu, R. Nishimaki, and K. Xagawa. Re-encryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices. In *Proc. of the17th International Conference on Practice and Theory in Public-Key Cryptography (PKC'14), Buenos Aires, Argentina*, volume 8383 of *Lecture Notes in Computer Science*, pages 95–112. Springer-Verlag, March 2014.

[12] N. Chandran, M. Chase, and V. Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In *Proc. of the 9th Theory of Cryptography Conference (TCC'12), Taormina, Sicily, Italy*, volume 7194 of *Lecture Notes in Computer Science*, pages 404–421. Springer-Verlag, March 2012.

[13] S. S. M. Chow and R. C. Phan. Proxy re-signatures in the standard model. In *Proc. of the 11th International Conference on Information Security (ISC'08), Taipei, Taiwan*, volume 5222 of *Lecture Notes in Computer Science*, pages 260–276. Springer-Verlag, September 2008.

[14] P. Dutta, W. Susilo, D. H. Duong, J. Baek, and P. S. Roy. Identity-based unidirectional proxy re-encryption in standard model: A lattice-based construction. In *Proc. of the The 21ST World Conference on Information Security Applications (WISA'20 ), Jeju, Korea*, pages 316–332. Springer, 2020.

[15] X. Fan and F. Liu. Proxy re-encryption and re-signatures from lattices. In *Proc. of the 17th International Conference on Applied Cryptography and Network Security (ACNS'19), Bogota, Colombia*, volume 11464 of *Lecture Notes in Computer Science*, pages 363–382. Springer-Verlag, June 2019.

[16] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[17] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of the 40th Annual ACM Symposium on Theory of Computing (STOC'08), Victoria, British Columbia, Canada*, pages 197–206. ACM, May 2008.

[18] M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Proc. of the 5th International Conference on Applied Cryptography and Network Security (ACNS'07), Zhuhai, China*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer-Verlag, June 2007.

[19] S. Hohenberger, G. N. Rothblum, A. Shelat, and V. Vaikuntanathan. Securely obfuscating re-encryption. In *Proc. of the 4th Theory of Cryptography Conference (TCC'07), Amsterdam, The Netherlands*, volume 4392 of *Lecture Notes in Computer Science*, pages 233–252. Springer-Verlag, February 2007.

[20] R. A. Horn, R. A. Horn, and C. R. Johnson. *Topics in matrix analysis*. Cambridge uni. press, 1994.

[21] J. Hou, M. Jiang, Y. Guo, and W. Song. Efficient identity-based multi-bit proxy re-encryption over lattice in the standard model. *Journal of Information Security and Applications*, 47:329–334, August 2019.

[22] E. Kirshanova. Proxy re-encryption from lattices. In *Proc. of the 17th International Conference on Practice and Theory in Public-Key Cryptography (PKC'14), Buenos Aires, Argentina*, volume 8383 of *Lecture Notes in Computer Science*, pages 77–94. Springer-Verlag, March 2014.

[23] B. Libert and D. Vergnaud. Multi-use unidirectional proxy re-signatures. In *Proc. of the 15th ACM Conference on Computer and Communications Security (CCS' 08), Alexandria, Virginia, USA*, pages 511–520. ACM, October 2008.

[24] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In *Proc. of the 11th International Conference on Practice and Theory in Public-Key Cryptography (PKC'08), Barcelona, Spain*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379. Springer-Verlag, March 2008.

[25] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'12), Cambridge, UK*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer-Verlag, April 2012.

[26] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *Proc. of the 33rd Annual Cryptology Conference (CRYPTO'13), Santa Barbara, CA, USA*, volume 8042 of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, August 2013.

[27] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *Proc. of the 45th Symposium on Foundations of Computer Science (FOCS'04), Rome, Italy*, pages 372–381. IEEE, October 2004.

[28] M. Miki, E. Hayashi, and H. Shingai. Highly reliable and highly secure online storage platform supporting "timeon" regza cloud service. *Toshiba Review*, 68(5):25–27, 2013.

[29] R. Nishimaki and K. Xagawa. Key-private proxy re-encryption from lattices, revisited. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 98(1):100–116, January 2015.

[30] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of the 37th Annual ACM Symposium on Theory of Computing (STOC'05), Baltimore, MD, USA*, pages 84–93. ACM, May 2005.

[31] J. Shao, Z. Cao, L. Wang, and X. Liang. Proxy re-signature schemes without random oracles. In *Proc. of the 8th International Conference on Cryptology in India ( INDOCRYPT'07), Chennai, India*, volume 4859 of *Lecture Notes in Computer Science*, pages 197–209. Springer-Verlag, December 2007.

[32] J. Shao, G. Wei, Y. Ling, and M. Xie. Unidirectional identity-based proxy re-signature. In *Proc. of the IEEE International Conference on Communications (ICC'11), Kyoto, Japan*, pages 1–5. IEEE, June 2011.

[33] K. Singh, C. P. Rangan, and A. K. Banerjee. Lattice based identity based proxy re-encryption scheme. *Journal of Internet Services and Information Security*, 3(3/4):38–51, November 2013.

[34] K. Singh, C. P. Rangan, and A. K. Banerjee. Lattice based identity based unidirectional proxy re-encryption scheme. In *Proc. of the 4th International Conference on Security, Privacy, and Applied Cryptography Engineering (SPACE'14), Pune, India*, volume 8804 of *Lecture Notes in Computer Science*, pages 76–91. Springer-Verlag, October 2014.

[35] T. Smith. Dvd jon: buy drm-less tracks from apple itunes (2005). `https://www.theregister.co.uk/2005/03/18/itunes_pymusique/` [Online; accessed on November 10, 2020].

[36] M. Tian. Identity-based proxy re-signatures from lattices. *Information Processing Letters*, 115(4):462–467, April 2015.

[37] D. K. Xagawa. *Cryptography with lattices*. PhD thesis, Stanford University, 2009.

[38] W. Yin, Q. Wen, W. Li, H. Zhang, and Z. P. Jin. Identity based proxy re-encryption scheme under LWE. *KSII Transactions on Internet & Information Systems*, 11(12), December 2017.

---

## Author Biography

**Priyanka Dutta** received her B.S. degree in Mathematics in 2012 from University of Calcutta, Kolkata, India. In 2014, she received her M.S. degree in Mathematics from Presidency University, Kolkata, India. She is currently a visiting fellow in the School of Computing and Information Technology, Faculty of Engineering and Information Sciences at the University of Wollongong (UOW), Australia. Her research interests include lattice-based post-quantum cryptographic protocols and their security.

**Willy Susilo** is a Senior Professor in the School of Computing and Information Technology, Faculty of Engineering and Information Sciences at the University of Wollongong (UOW), Australia. He is the director of Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, UOW and the Head of School of Computing and Information Technology at UOW (2015 - now). Prior to this role,he was awarded the prestigious Australian Research Council Future Fellowship in 2009. In 2016, he was awarded the "Researcher of the Year" at UOW, due to his research excellence and contributions. He is the Editor-in-Chief of the Elsevier's Computers Standards and Interface and the Information journal. He is currently an Associate Editor of IEEE Transactions on Dependable and Secure Computing. He has also served as the program committee member of several international conferences.

**Dung Hoang Duong** is a lecturer in the School of Computing and Information Technology at University of Wollongong since 2018. He received a Ph.D. degree in mathematics from Leiden University, Netherlands, in 2013. He was a postdoctoral fellow in the Faculty of Mathematics at Bielefeld University from 2013 to 2015. From 2015 to 2018, he was an assistant professor at the Institute of Mathematics for Industry at Kyushu University, Japan. His research interests are post-quantum cryptography,especially lattice-based cryptography and multivariate public key cryptography.

**Joonsang Baek** received the Ph.D. degree from Monash University, Australia, in 2004. His Ph.D. thesis was on security analysis of signcryption, and has received great attention from the research community. He was a Research Scientist in the Institute for Infocomm Research, Singapore, and an Assistant Professor in the Khalifa University of Science and Technology, United Arab Emirates. He is currently a Senior Lecturer in the School of Computer Science and Information Technology, University of Wollongong, Australia. He has published his work in numerous reputable journals and conference proceedings. His current research interests are in the field of applied cryptography and cybersecurity. He has also served as a Program Committee Member and the Chair for a number of renowned conferences on information security and cryptography.

**Partha Sarathi Roy** is a lecturer in the School of Computing and Information Technology at University of Wollongong since 2019. He received a Ph.D. degree in mathematics from University of Calcutta, India, in 2015. He was an assistant professor at the Department of Informatics at Kyushu University from 2016 to 2017. From 2017 to 2019, he was a research engineer at KDDI Research, Japan. His research interests are Lattice-Based Cryptography, Code-Based Cryptography, Secret Sharing Schemes, Oblivious Transfer Protocols, Secure Composition (Provable Security).