Lattice-Based Identity-Based Accountable Proxy Re-Encryption Scheme Postprint

Authors: Meng Hui, Ren Lina, Li Ying

Date: 2022-04-07T15:01:56Z

Abstract

To address issues such as key abuse and digital certificate management in existing lattice-based proxy re-encryption schemes, we introduce an accountability mechanism and propose a novel identity-based accountable proxy re-encryption scheme. This scheme employs user identity ID to compute and generate a matrix as the public key, and utilizes the preimage sampling algorithm to extract the private key, thereby resolving the problem of digital certificate management. It leverages both users' public keys to compute and generate the re-encryption key, which enhances computational efficiency during encryption and decryption. Furthermore, it incorporates the proxy's public-private key pair in the re-encryption operation to implement the accountability algorithm, effectively mitigating collusion between the proxy and the delegatee. Security analysis demonstrates that the scheme achieves security against chosen-plaintext attacks; regarding efficiency, the scheme exhibits relatively low computational complexity and ciphertext overhead.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers

Identity-based Accountable Proxy Re-encryption Scheme from Lattices

Meng Hui†, Ren Lina, Li Ying
(School of Computer Science & Technology, Henan Polytechnic University, Jiaozuo, Henan 454003, China)

Abstract: To address key abuse and digital certificate management issues in existing lattice-based proxy re-encryption schemes, this paper introduces an accountability mechanism and proposes a novel identity-based accountable proxy re-encryption scheme. The scheme computes a matrix from the user' s identity ID to serve as the public key and employs a preimage sampling algorithm to extract private keys, thereby solving the digital certificate management problem. It uses both users' public keys to compute and generate re-encryption keys, improving computational efficiency during encryption and decryption. The proxy' s public and private keys participate in re-encryption operations to complete the accountability algorithm, effectively curbing collusion between the proxy and the delegatee. Security analysis demonstrates that the scheme satisfies chosen-plaintext attack security, while efficiency analysis shows low computational complexity and ciphertext overhead.

Keywords: proxy re-encryption; lattice; learning with error; accountability

0 Introduction

Cloud storage and data sharing currently occupy a central position in network data storage and computation. Users store large amounts of data in network cloud drives to reduce the burden on their own storage devices while facilitating data sharing among users. In complex network environments, users need to encrypt data before uploading it to cloud servers for storage or sharing to protect data privacy. However, data senders must constantly monitor whether users are accessing the data and download the requested data before forwarding it to recipients. Proxy re-encryption solves the problem of data owners needing to be continuously online in traditional cloud computing environments, reduces the burden of frequently accessing cloud ciphertext data, and enhances data reliability and confidentiality.

In 1998, Blaze et al. [1] first introduced the concept of Proxy Re-encryption (PRE), adding a proxy role to public key encryption systems where the proxy uses re-encryption keys to perform ciphertext transformation. In 2007, Green et al. [2] proposed the first Identity-Based Proxy Re-encryption (IB-PRE) scheme, where user identity ID serves directly as the public key, solving the certificate management problem in public key infrastructure. This scheme satisfies multi-hop and non-interactive properties. However, with the development of quantum computers, the security of traditional number-theoretic hard problems is threatened. In 2010, Xagawa et al. [3] first proposed a lattice-based proxy re-encryption scheme that not only resists quantum attacks but also reduces computational complexity. In 2014, Singh et al. [4] integrated identity-based encryption with proxy re-encryption, proposing a PRE scheme that can encrypt multi-bit information and improve operational efficiency. Their scheme satisfies anonymity and multi-hop properties. In 2021, Tang et al. [5] constructed a PRE scheme using the RLWE (Ring Learning With Errors) problem, effectively reducing ciphertext and key sizes while improving encryption and decryption efficiency. However, all these schemes are bidirectional, cannot resist collusion attacks, and suffer from key leakage and other security issues.

In 2016, Kim et al. [6] proposed the first unidirectional proxy re-encryption scheme based on worst-case lattice hard problems, where the delegatee cannot perceive the proxy' s existence—ciphertexts re-encrypted using the re-encryption key are indistinguishable from those encrypted directly with the delegatee' s public key. In 2020, Wang et al. [7] identified issues in Kim' s scheme where re-encrypted ciphertexts could not be decrypted or had high error rates, proposing a new unidirectional proxy re-encryption scheme proven to satisfy chosen-plaintext attack (CPA) security. In 2021, Dutta et al. [8] presented the first concrete construction of a collusion-resistant unidirectional IB-PRE for selective and adaptive identities. Their construction is non-interactive and non-transferable but does not satisfy multi-hop properties.

In 2013, Wang et al. [9] proposed a new primitive called PRE+, which differs from traditional PRE in terms of who delegates decryption rights. In traditional PRE, the delegator is the ciphertext recipient, whereas in PRE+, the delegator is the ciphertext sender—i.e., the encryptor delegates decryption rights to the delegatee. Additionally, the input elements in the re-encryption key generation algorithm differ: traditional PRE generates re-encryption keys using the recipient' s private key and the delegatee' s private or public key, while PRE+ uses only the public keys of both parties. In 2020, Singh et al. [10] proposed unidirectional PRE and PRE+ schemes. PRE+ improves computational efficiency during encryption and decryption, making it suitable for fine-grained and non-transferable authorization, with broad applications in secure cloud computing and multicasting. Although collusion attacks between the delegatee and proxy do not expose the delegator' s long-term key, they may collude to generate new re-encryption keys for malicious users not trusted by the delegator. Moreover, the inherent functionality of proxy re-encryption may lead to re-encryption key abuse issues, where the proxy and delegatee collude to obtain the delegator' s decryption capability and store it on any carrier, such as a decryption device.

To mitigate this problem, Ateniese et al. [11] introduced the concept of non-transferability in 2005: when Bob and the proxy collude to distribute Alice' s decryption capability, Bob must publicly expose his own decryption capability as a penalty. In 2019, Guo et al. [12] constructed a non-transferable proxy re-encryption scheme using indistinguishability obfuscation and K-unforgeable authentication. While non-transferability deters malicious users to some extent, when Bob' s key is far less valuable than the data owner' s key, Bob might expose his decryption capability for greater benefit. In this case, the proxy can distribute Alice' s decryption capability without any cost and even deny its malicious behavior.

In 2021, Guo et al. [13] addressed these issues by proposing an Accountable Proxy Re-encryption (APRE) scheme that introduces a judgment algorithm to determine whether the proxy is denying its behavior of distributing the delegator' s decryption capability. The accountable proxy re-encryption model is illustrated in Figure 1.

1 Preliminaries

1.1 Lattices

Let $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n\}$ be a matrix composed of n linearly independent vectors in $\mathbb{R}^m$. The set of all integer linear combinations of these n vectors forms an m-dimensional lattice $\mathcal{L}$, i.e.:

$$\mathcal{L} = \{\mathbf{y} = \sum_{i=1}^{n} s_i \mathbf{b}_i \mid s_i \in \mathbb{Z}\}$$

We call B a basis of lattice $\mathcal{L}$. When $n = m$, the lattice $\mathcal{L}$ is called a full-rank lattice.

Let q be a prime number and $A \in \mathbb{Z}_q^{n \times m}$. Define two m-dimensional full-rank lattices:

$$\Lambda_q^{\perp}(A) = \{\mathbf{e} \in \mathbb{Z}^m \mid A\mathbf{e} = \mathbf{0} \pmod q\}$$

$$\Lambda_q(A) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{y} = A^T\mathbf{s} \pmod q\}$$

Trapdoor functions are widely used in lattice-based cryptography. A trapdoor basis T is a short basis of a lattice, and the scheme uses a trapdoor generation algorithm to produce the trapdoor basis as the master secret key.

Definition 1. Let q be a prime number and $A \in \mathbb{Z}_q^{n \times m}$. When $n \geq 1$, the lattice $\Lambda_q^{\perp}(A)$ is a full-rank lattice.

Definition 2. Let q be a prime number and matrix $A \in \mathbb{Z}_q^{n \times m}$. Define the m-dimensional lattice:

$$\Lambda_q^{\perp}(A) = \{\mathbf{e} \in \mathbb{Z}^m \mid A\mathbf{e} = \mathbf{0} \pmod q\}$$

1.2 Discrete Gaussian Distribution

Gaussian distributions are commonly used in lattice hard problem research. This section provides a detailed introduction to discrete Gaussian distributions and related lemmas.

For any vector $\mathbf{c} \in \mathbb{R}^m$ and real number $\sigma > 0$, define the m-dimensional lattice Λ with discrete Gaussian distribution:

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} = \frac{e^{-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2}}{\sum_{\mathbf{x}\in\Lambda} e^{-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2}}$$

where $\mathbf{x} \in \Lambda$.

Literature [14] introduced the preimage sampling algorithm, which includes the SamplePre algorithm that uses a trapdoor basis T to solve for the preimage $\mathbf{x}$ corresponding to a given image value $\mathbf{u}$.

Lemma 1 [14]. Trapdoor generation algorithm: For integer $n \geq 1$ and $q \geq 3$, there exists a PPT algorithm $\mathrm{TrapGen}(q, n)$ that outputs a matrix $A \in \mathbb{Z}_q^{n\times m}$ and a trapdoor basis $T \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(A)$, where $m = \lceil 6n \log q \rceil$, $\|\tilde{T}\| \leq O(\sqrt{n \log q})$, and the distribution of A is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$.

Lemma 2 [14]. Let $q \geq 3$ be an integer, $A \in \mathbb{Z}_q^{n\times m}$ with a trapdoor basis $T \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^{\perp}(A)$, and real number $\sigma \geq \|\tilde{T}\| \cdot \omega(\sqrt{\log n})$. There exists a PPT algorithm $\mathrm{SamplePre}(A, T, \mathbf{u}, \sigma)$ that, for any vector $\mathbf{u} \in \mathbb{Z}_q^n$, samples a vector $\mathbf{e} \in D_{\Lambda_q^{\perp}(A),\sigma}$ such that $A\mathbf{e} = \mathbf{u} \pmod q$.

Lemma 3 [15]. Let positive integers n, m, prime q, and real $\sigma \geq \omega(\sqrt{\log m})$. For $A \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, the distribution of $\mathbf{e}$ output by $\mathrm{SamplePre}(A, T, \mathbf{u}, \sigma)$ is statistically close to $D_{\mathbb{Z}^m,\sigma}$.

1.3 Lattice Hard Problems

Definition 2 [16]. Small Integer Solution Problem (SIS). Given a prime q, matrix $A \in \mathbb{Z}_q^{n\times m}$, and constant β, find a non-zero vector $\mathbf{z} \in \mathbb{Z}^m$ such that $A\mathbf{z} = \mathbf{0} \pmod q$ and $\|\mathbf{z}\|_\infty \leq \beta$.

Literature [17] provides a reduction from worst-case hardness to SIS, proving that the SIS problem is sufficiently hard for certain parameters.

Theorem 1 [17]. Let integer $n \geq 1$, $q \geq 2$, $m = \mathrm{poly}(n)$, and $\beta \geq \max\{1, \sqrt{n \log q}\} \cdot \omega(\sqrt{\log m})$. Then solving the SIS problem with solution set $\{\mathbf{z} \in \mathbb{Z}^m \mid \|\mathbf{z}\|_\infty \leq \beta\}$ is at least as hard as solving the worst-case lattice problem in n-dimensional lattices.

Definition 3 [18]. Learning With Errors Problem (LWE). Given positive integers n and q, a uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$, and an error distribution χ over $\mathbb{Z}_q$, the LWE problem is to find $\mathbf{s}$ with non-negligible probability given $(A, A^T\mathbf{s} + \mathbf{e})$ where $\mathbf{e} \leftarrow \chi$.

Definition 4 [4]. Decisional LWE Problem (DLWE). Let positive integers n, m, prime q, and $\alpha \in (0, 1)$ with $2\sqrt{n} \leq q\alpha$. Let χ be a Gaussian distribution over $\mathbb{Z}_q$. The decisional LWE problem is to distinguish, given polynomially many independent samples, whether they are from distribution $(A, A^T\mathbf{s} + \mathbf{e})$ or from the uniform random distribution over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$.

Literature [18] establishes the relationship between LWE and DLWE and provides a reduction from the Shortest Vector Problem (SVP) to LWE, proving that LWE is sufficiently hard for certain parameters.

Theorem 2 [18]. Let n, q be positive integers and $\alpha \in (0, 1)$ with $2\sqrt{n} \leq q\alpha$. If the LWE problem can be solved by an efficient algorithm, then the worst-case Gap-SVP and SIVP problems in n-dimensional lattices can also be solved by a quantum algorithm with time complexity $\widetilde{O}(n/\alpha)$.

---

2 Definitions

2.1 Definition of Identity-Based Accountable Proxy Re-encryption

This section introduces the definition of identity-based accountable proxy re-encryption. Let λ be the security parameter. The scheme consists of the following algorithms:

1. **Setup**(λ): Takes security parameter λ as input and outputs public parameters *params* and master secret key *msk*.
2. **Extract**$(params, msk, id)$: Takes public parameters *params*, master secret key *msk*, and user identity *id* as input, and outputs user secret key SK_{id}.
3. **Enc**$(params, id_i, m)$: Takes public parameters *params*, user identity id_i, and plaintext m as input, and outputs second-level ciphertext C_i.
4. **ReKeyGen**$(params, id_i, id_j, id_p)$: Takes public parameters *params*, identities of both users and the proxy as input, and outputs re-encryption key $rk_{i\to j}$.
5. **ReEnc**$(params, rk_{i\to j}, SK_p, C_i)$: Takes public parameters *params*, re-encryption key $rk_{i\to j}$, proxy secret key SK_p, and second-level ciphertext C_i as input, and outputs first-level ciphertext C_j.
6. **Dec2**$(params, SK_i, C_i)$: Takes public parameters *params*, secret key SK_i, and second-level ciphertext C_i as input, and decrypts to recover plaintext m.
7. **Dec1**$(params, SK_j, C_j)$: Takes public parameters *params*, secret key SK_j, and first-level ciphertext C_j as input, and outputs plaintext m.
8. **Judge**$(params, id_i, id_p, \mu, D)$: Takes public parameters *params*, identities id_i and id_p, probability parameter μ, and a decryption device D as input. Through black-box access to the decryption device, this algorithm outputs either "Proxy" or "Delegator" , identifying the creator of the malicious decryption device.

Definition 5. IND-CPA Security. The advantage of adversary $\mathcal{A}$ is defined as:

$$\mathrm{ADV}_{\mathrm{PRE},\mathcal{A}}^{\text{IND-CPA}} = \left|\Pr[\mathrm{Exp}_{\mathrm{PRE},\mathcal{A}}^{\text{IND-CPA}} = 1] - \frac{1}{2}\right|$$

When the advantage of all PPT adversaries $\mathcal{A}$ is negligible, the scheme is said to be IND-CPA secure.

Definition 6. Malicious Proxy Security. The advantage of adversary $\mathcal{A}$ is defined as:

$$\mathrm{ADV}^{\mathrm{mp}}_{\mathrm{PRE},\mathcal{A}} = \left|\Pr[\mathrm{Exp}^{\mathrm{mp}}_{\mathrm{PRE},\mathcal{A}} = 1]\right|$$

When the advantage of all PPT adversaries $\mathcal{A}$ is negligible, the scheme satisfies malicious proxy security.

Definition 7. Malicious Delegator Security. The advantage of adversary $\mathcal{A}$ is defined as:

$$\mathrm{ADV}^{\mathrm{md}}_{\mathrm{PRE},\mathcal{A}} = \left|\Pr[\mathrm{Exp}^{\mathrm{md}}_{\mathrm{PRE},\mathcal{A}} = 1]\right|$$

When the advantage of all PPT adversaries $\mathcal{A}$ is negligible, the scheme is said to be malicious delegator secure.

Correctness: The scheme correctly recovers the plaintext if it satisfies the following two conditions:

1. If $C_i \leftarrow \mathrm{Enc}(params, id_i, m)$, then $\mathrm{Dec2}(params, SK_i, C_i) = m$.
2. If $C_j \leftarrow \mathrm{ReEnc}(params, \mathrm{ReKeyGen}(params, id_i, id_j, id_p), SK_p, \mathrm{Enc}(params, id_i, m))$, then $\mathrm{Dec1}(params, SK_j, C_j) = m$.

If a PRE scheme simultaneously satisfies malicious proxy security and malicious delegator security, it is said to satisfy accountability.

---

3 Construction

3.1 Scheme Construction

This section presents the accountable proxy re-encryption security model based on the IND-aID-CPA game [19]. Let λ be the security parameter. The game consists of adversary $\mathcal{A}$ and the following oracles:

Initialization Phase: Challenger $\mathcal{C}$ runs $\mathrm{Setup}(\lambda)$ and sends the public parameters *params* to adversary $\mathcal{A}$.

Phase 1: Adversary $\mathcal{A}$ issues queries, and challenger $\mathcal{C}$ responds:

- **Secret key generation oracle $\mathcal{O}_{\mathbf{Extract}}$:** When adversary $\mathcal{A}$ inputs id_i, if user i is malicious, challenger $\mathcal{C}$ sends SK_i to $\mathcal{A}$; otherwise, it sends $\perp$.
- **Re-encryption key generation oracle $\mathcal{O}_{\mathbf{ReKeyGen}}$:** Adversary $\mathcal{A}$ inputs (id_i, id_j, id_p). Challenger $\mathcal{C}$ computes and sends the re-encryption key $rk_{i\to j}$ to $\mathcal{A}$.

- **Re-encryption oracle $\mathcal{O}_{\textbf{ReEnc}}$:** If user j is dishonest, adversary $\mathcal{A}$ inputs (id_i, id_j, id_p, C_i) and challenger $\mathcal{C}$ outputs $\perp$; otherwise, it outputs the re-encrypted ciphertext C_j.
- **Challenge oracle $\mathcal{O}_{\textbf{Challenge}}$:** Adversary $\mathcal{A}$ inputs a target user and two messages m_0, m_1. The oracle randomly selects $b \in \{0,1\}$ and returns the challenge ciphertext $C^* = \mathrm{Enc}(params, id^*, m_b)$ to $\mathcal{A}$.

Guess: Adversary $\mathcal{A}$ inputs a guess $b' \in \{0,1\}$. If $b' = b$, challenger $\mathcal{C}$ outputs 1; otherwise, it outputs 0.

Based on the accountability algorithm and LWE hard problem, this paper combines proxy re-encryption with identity-based encryption to propose an identity-based accountable proxy re-encryption scheme from lattices. The concrete construction is as follows:

(1) Setup(λ): Let λ be the security parameter. Generate a random matrix $A \in \mathbb{Z}_q^{n\times m}$ and a trapdoor basis $T \in \mathbb{Z}_q^{m\times m}$ using the trapdoor generation algorithm $\mathrm{TrapGen}(q, n)$. Randomly select $m+1$ matrices $U_0, U_1, ..., U_m \in \mathbb{Z}_q^{n\times m}$ that are linearly independent. The master public key is $mpk = (A, U_0, U_1, ..., U_m)$, the master secret key is $msk = T$, and the public parameters are $params = (m, n, q, mpk)$.

(2) Extract($params, msk, id$): Input public parameters $params$, master secret key msk, and user/proxy identity id. Compute $\mathbf{u}_i = \sum_{k=0}^{m} id_i^k U_k \pmod q$. Use the preimage sampling algorithm to generate a vector $\mathbf{x}_i$ such that $A\mathbf{x}_i = \mathbf{u}_i \pmod q$ and $\|\mathbf{x}_i\| \le \sigma$. The user/proxy secret key is $SK_{id} = \mathbf{x}_i$.

(3) Enc($params, id_i, m$): Input user i' s identity id_i and plaintext $m \in \{0,1\}$. Randomly select vector $\mathbf{s} \in \mathbb{Z}_q^n$ and error vector $\mathbf{e} \leftarrow \chi^m$. Compute:

$$C_1 = U_{id_i}^T \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$$

$$C_2 = \mathbf{t}^T \mathbf{s} + e' + m\lfloor q/2 \rfloor \in \mathbb{Z}_q$$

where $U_{id_i} = \sum_{k=0}^{m} id_i^k U_k$. Output second-level ciphertext $C = (C_1, C_2)$.

(4) ReKeyGen($params, mpk, id_i, id_j, id_p$): Input user identities id_i, id_j and proxy identity id_p. Select random vector $\mathbf{s}' \in \mathbb{Z}_q^n$ and error vector $\mathbf{e}_1 \leftarrow \chi^m$. Compute:

$$rk_{i\to j} = -\mathbf{x}_i^T U_{id_j} + \mathbf{x}_p^T U_{id_i} + \mathbf{e}_1 \in \mathbb{Z}_q^m$$

where $\mathbf{x}_i, \mathbf{x}_p$ are the secret keys of user i and proxy p, respectively.

(5) ReEnc($params, rk_{i\to j}, SK_p, C_i$): Input re-encryption key $rk_{i\to j}$, proxy secret key SK_p, and second-level ciphertext $C_i = (C_{i,1}, C_{i,2})$. Select random error vector $\mathbf{e}_2 \leftarrow \chi^m$ and compute:

$$C_{j,1} = C_{i,1} + rk_{i\to j}^T + \mathbf{e}_2 \in \mathbb{Z}_q^m$$

$$C_{j,2} = C_{i,2} \in \mathbb{Z}_q$$

Output first-level ciphertext $C_j = (C_{j,1}, C_{j,2})$.

(6) Dec2($params, SK_i, C_i$**):** Input secret key $SK_i = \mathbf{x}_i$ and second-level ciphertext $C_i = (C_{i,1}, C_{i,2})$. Compute:

$$\mu = C_{i,2} - \mathbf{x}_i^T C_{i,1} \pmod q$$

If μ is closer to 0, output $m = 0$; otherwise, output $m = 1$.

(7) Dec1($params, SK_j, C_j$**):** Input secret key $SK_j = \mathbf{x}_j$ and first-level ciphertext $C_j = (C_{j,1}, C_{j,2})$. Compute:

$$\mu' = C_{j,2} - \mathbf{x}_j^T C_{j,1} \pmod q$$

If μ' is closer to 0, output $m = 0$; otherwise, output $m = 1$.

(8) Judge($params, id_i, id_p, \mu, D$**):** Provide a decryption device D as an oracle:

1. Repeat the following experiment $n/\lambda\mu$ times: Uniformly randomly select vector $\mathbf{s} \in \mathbb{Z}_q^n$ and plaintext m, compute $C \leftarrow \text{Enc}(params, id_i, m)$; run decryption device D with input C and output m'.
2. If the majority of outputs are correct, output "Proxy" and exit; otherwise output "Delegator" .

3.2 Correctness

(1) Second-level ciphertext decryption: Since $\mathbf{e}$ and $\mathbf{e}'$ are selected from the Gaussian distribution set χ, when $\|\mathbf{e}\|_\infty < q/4$, the plaintext m can be successfully recovered.

(2) First-level ciphertext decryption: Because $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}, \mathbf{e}'$ are selected from the Gaussian distribution set χ, when $\|\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}\|_\infty < q/4$, the plaintext m can be successfully recovered.

3.3 Multi-hop Property

The proxy performs the first re-encryption:

$$C_{j,1} = \mathbf{y}^T U_{id_i} + \mathbf{e} + rk_{i\to j}^T + \mathbf{e}_2$$

$$= \mathbf{y}^T U_{id_j} + (\mathbf{x}_p^T U_{id_i} + \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e})$$

The proxy performs the second re-encryption:

$$C_{k,1} = C_{j,1} + rk_{j\to k}^T + \mathbf{e}_3$$

$$= \mathbf{y}^T U_{id_k} + (\mathbf{x}_p^T U_{id_i} + \mathbf{x}_p^T U_{id_j} + \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e})$$

After $N-1$ re-encryptions, using user N' s secret key to decrypt:

$$C_{N,1} = \mathbf{y}^T U_{id_N} + \left(\sum_{i=1}^{N-1} \mathbf{x}_p^T U_{id_i} + \sum_{i=1}^{N} \mathbf{e}_i + \mathbf{e} \right)$$

The accumulated error term satisfies $\| \sum_{i=1}^{N} \mathbf{e}_i + \mathbf{e} \|_\infty < q/4$, allowing successful decryption.

---

4 Security and Efficiency Analysis

4.1 CPA Security

Theorem 4.1. Let λ be the security parameter. For any plaintext $m \in \{0,1\}$, if the LWE problem is unsolvable in polynomial time, then the scheme satisfies IND-aID-CPA security. That is, the advantage of adversary $\mathcal{A}$ in breaking the scheme within polynomial time t is negligible.

Proof: We prove that the advantage of PPT adversary $\mathcal{A}$ in successfully attacking the scheme is negligible by demonstrating the indistinguishability of the following games.

Game 1: The original IND-aID-CPA scheme. In the challenge phase, when challenger $\mathcal{C}$ receives the challenge identity id^* and messages m_0, m_1, it randomly selects one plaintext m_b and computes the challenge ciphertext $C^* = (C_1^*, C_2^*)$ where:

$$C_1^* = U_{id^*}^T \mathbf{s} + \mathbf{e}, \quad C_2^* = \mathbf{t}^T \mathbf{s} + e' + m_b \lfloor q/2 \rfloor$$

Finally, adversary $\mathcal{A}$ guesses b'. If successful, $\mathcal{C}$ outputs 1; otherwise, it outputs 0.

Game 2: In Game 2, matrix A is generated differently. Challenger $\mathcal{C}$ simulates the real scheme and answers adversary $\mathcal{A}$' s queries. Initially, $\mathcal{C}$ selects a random matrix $A \in \mathbb{Z}_q^{n\times m}$ and $m+1$ matrices $U_0, U_1, ..., U_m \in \mathbb{Z}_q^{n\times m}$

drawn from Gaussian distribution $D_{\mathbb{Z},\sigma}$. If the generated matrices are linearly dependent, they are regenerated. Challenger $\mathcal{C}$ sends public parameters $params = (A, U_0, U_1, ..., U_m)$ to $\mathcal{A}$.

Phase 1: Adversary $\mathcal{A}$ can make the following queries:

- **Proxy key query:** Challenger $\mathcal{C}$ computes the proxy' s secret key SK_p using the trapdoor T and sends it to $\mathcal{A}$.
- **Key extraction query:** Adversary $\mathcal{A}$ sends user identity id to challenger $\mathcal{C}$, which computes SK_{id} using the trapdoor and sends it to $\mathcal{A}$.
- **Re-encryption key query:** Adversary $\mathcal{A}$ sends identity set (id_i, id_j, id_p) to challenger $\mathcal{C}$, which computes $U_{id_i}, U_{id_j}, U_{id_p}$ and generates the re-encryption key $rk_{i\to j}$ using the computed public keys.

Phase 2 (Challenge): When challenger $\mathcal{C}$ receives the challenge identity id^* and messages m_0, m_1 from adversary $\mathcal{A}$, it randomly selects $b \in \{0, 1\}$ and sends the challenge ciphertext $C^* = (C_1^*, C_2^*)$ to $\mathcal{A}$, where $C_1^* = U_{id^*}^T \mathbf{s} + \mathbf{e}$ and $C_2^* = \mathbf{t}^T\mathbf{s} + e' + m_b\lfloor q/2 \rfloor$. Finally, adversary $\mathcal{A}$ guesses b'. If $b' = b$, challenger $\mathcal{C}$ outputs 1; otherwise, it outputs 0.

Game 3: In Game 2, the challenge ciphertext is computed through the encryption algorithm. In Game 3, C_1^* and C_2^* are directly chosen uniformly at random from $\mathbb{Z}_q^m \times \mathbb{Z}_q$. Clearly, adversary $\mathcal{A}$ obtains advantage 0 in Game 3.

We prove the indistinguishability of the two games by reducing to the LWE hard problem. If adversary $\mathcal{A}$ can distinguish the two games with advantage ε, then algorithm $\mathcal{B}$ can break the LWE hard problem [12].

Algorithm $\mathcal{B}$: Upon receiving random instance $(A, \mathbf{y}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, algorithm $\mathcal{B}$ sets $C_1^* = \mathbf{y}$ and $C_2^* = \mathbf{t}^T\mathbf{s} + e' + m_b\lfloor q/2 \rfloor$. If adversary $\mathcal{A}$ successfully guesses m, algorithm $\mathcal{B}$ outputs 1; otherwise, it outputs 0.

If $\mathbf{y}$ is uniformly random, then C_1^* is also uniformly random, and adversary $\mathcal{A}$ succeeds with probability at most 1/2. If $\mathbf{y}$ is generated as $A^T\mathbf{s} + \mathbf{e}$, then C_1^* follows the correct distribution, and algorithm $\mathcal{B}$ solves the LWE problem with advantage $\varepsilon/2$. However, since LWE is a lattice hard problem, algorithm $\mathcal{B}$ cannot solve it, making ε negligible.

Thus, Games 2 and 3 are indistinguishable, and adversary $\mathcal{A}$ obtains advantage 0 in Game 2. Similarly, Games 1 and 2 are indistinguishable because matrix A is random and U_i follow Gaussian distributions. By Theorem 3, the distribution of U_{id} is statistically close to uniform, making Games 1 and 2 indistinguishable. Therefore, adversary $\mathcal{A}$ also obtains advantage 0 in Game 1.

In conclusion, the APRE scheme satisfies IND-aID-CPA security in the standard model.

4.2 Accountability

To prove the scheme' s accountability, we demonstrate that it satisfies both malicious proxy security and malicious delegator security.

4.2.1 Malicious Proxy Security **Theorem 4.2.** Under the LWE hard problem, if adversary $\mathcal{A}$' s advantage in successfully framing the delegator within polynomial time t is negligible, then the scheme satisfies malicious proxy security. That is, the advantage of adversary $\mathcal{A}$ in outputting a decryption device that causes the judge algorithm to output "Delegator" is negligible.

Proof: We prove malicious proxy security through two experiments, showing their computational indistinguishability and that the adversary' s advantage in the experiment is negligible.

Experiment $\mathrm{Exp}_0^{\mathbf{mp}}$ (Original malicious proxy security experiment): Adversary $\mathcal{A}$ outputs a decryption device D that can obtain plaintext with non-negligible probability μ. Challenger $\mathcal{C}$ runs decryption device D with n irregular plaintexts as input. Irregular ciphertexts are generated as $C = (C_1, C_2)$ where $C_1 = U_{id_i}^T \mathbf{s} + \mathbf{e}$ and $C_2 = \mathbf{t}^T \mathbf{s} + e' + m\lfloor q/2 \rfloor$.

Experiment $\mathrm{Exp}_1^{\mathbf{mp}}$ (Modified malicious proxy security experiment): Differs from $\mathrm{Exp}_0^{\mathrm{mp}}$ only in that the ciphertexts are generated as $C_1 = \mathbf{r}^T U_{id_i} + \mathbf{e}$ where $\mathbf{r}$ is uniformly random. Since $\mathbf{r}$ is uniform, the distribution of C_1 is statistically close to uniform over $\mathbb{Z}_q^m$, making the experiments indistinguishable.

Lemma 4.1. Under the LWE hard problem assumption, $\mathrm{Exp}_0^{\mathrm{mp}}$ and $\mathrm{Exp}_1^{\mathrm{mp}}$ are computationally indistinguishable.

Lemma 4.2. In $\mathrm{Exp}_1^{\mathrm{mp}}$, the advantage of adversary $\mathcal{A}$ in outputting a decryption device that causes the judge algorithm to output "Delegator" is negligible. In one trial, since the input ciphertext is a regular ciphertext, D returns correct plaintext m with probability μ. The adversary' s advantage in $\mathrm{Exp}_1^{\mathrm{mp}}$ is $(1/\lambda)(1-\mu)^{n/\lambda\mu} \leq e^{-n/\lambda}$, which is negligible.

4.2.2 Malicious Delegator Security **Theorem 4.3.** Under the LWE hard assumption, if adversary $\mathcal{A}$' s advantage in successfully framing the proxy within polynomial time t is negligible, then the scheme satisfies malicious delegator security. That is, the advantage of adversary $\mathcal{A}$ in outputting a decryption device that causes the judge algorithm to output "Proxy" is negligible.

Proof: We prove malicious delegator security through two experiments.

Experiment $\mathrm{Exp}_0^{\mathbf{md}}$ (Original malicious delegator security experiment): When adversary $\mathcal{A}$ outputs a decryption device D that can obtain plaintext with non-negligible probability μ, challenger $\mathcal{C}$ runs D with n irregular plaintexts as input. Irregular ciphertexts are generated as in the real scheme.

Experiment $\mathrm{Exp}_1^{\mathbf{md}}$ (Modified malicious delegator security experiment): Differs from $\mathrm{Exp}_0^{\mathrm{md}}$ only in that the ciphertexts are generated using uniformly random matrix $R \in \mathbb{Z}_q^{n\times m}$. Since R is uniform, the distribution is statistically close to uniform, making the experiments indistinguishable.

Lemma 4.3. Under the LWE hard problem assumption, $\mathrm{Exp}_0^{\mathrm{md}}$ and $\mathrm{Exp}_1^{\mathrm{md}}$ are computationally indistinguishable.

Lemma 4.4. In $\mathrm{Exp}_1^{\mathrm{md}}$, the advantage of adversary $\mathcal{A}$ in outputting a decryption device that causes the judge algorithm to output "Proxy" is negligible. The decryption device' s input ciphertext is $C = (C_1, C_2)$ where $C_1 = R^T\mathbf{s}+\mathbf{e}$. Since R is uniform, the adversary cannot recover plaintext m, making the advantage negligible.

4.3 Efficiency Analysis

Literature [13] proposes an accountable proxy re-encryption scheme based on the DBDH hard assumption, which has high computational complexity and certificate management issues. Literature [20] presents a lattice-based identity-based PRE scheme that can encrypt multi-bit information, but it is interactive and requires the delegatee to provide their private key for re-encryption key generation, leading to potential key leakage. Literature [21] proposes a single-hop lattice-based PRE scheme using trapdoors for secret key generation. Our scheme achieves accountability and collusion resistance, provides unidirectionality and multi-hop properties with good ciphertext expansion, and resists quantum attacks.

Table 1 compares storage space and performance among literature [13], [20], and our scheme, where $|G|$ is the number of elements in group G, $|p|$ denotes the bit length of elements in $\mathbb{Z}_p$, and $M = 2^m$.

Table 2 compares computational and communication complexity among three lattice-based proxy re-encryption schemes: literature [20], [21], and our scheme. Ciphertext overhead refers to the ciphertext size per 1-bit plaintext. MMM denotes matrix-matrix multiplication, MMA matrix-matrix addition, VMM vector-matrix multiplication, VCM constant-vector multiplication, VVM vector-vector addition, EGT exponentiation in group G_T, and EG exponentiation in group G.

Table 1. Comparison of Storage Space and Performance

Scheme	Hard Problem	Unidirectionality	IBE	Accountability
[13]	DBDH	No	No	Yes
[20]	LWE	No	Yes	No
[21]	LWE	Yes	No	No
Ours	LWE	Yes	Yes	Yes

Table 2. Comparison of Computational and Communication Complexity

Scheme	Encryption	Re-encryption	Decryption	Ciphertext Overhead
[13]	2EGT+2EG	2EGT+2EG	2EG	$2\|G_T\| + 2\|G\|$
[20]	2VMM+2VCM+3VVA	2VMM+2VCM+3VVA	VMM+VCM+2VVA	$(n+m)\log q$
[21]	2MMM+2MMA+1MMM	2MMM+2MMA+2VMM	1VMM+1VCM+3VVA	$2n \log q$
Ours	2VMM+1VCM+3VVA	2VMM+1VCM+3VVA	VMM+VCM+2VVA	$(n+1)\log q$

The comparison results show that our scheme has smaller secret key sizes than literature [21], satisfies unidirectionality and accountability compared to other schemes, has lower computational complexity than literature [20], and maintains lower key overhead compared to literature [13] and [21]. Therefore, our scheme offers good security and high efficiency.

5 Conclusion

This paper proposes an identity-based accountable proxy re-encryption scheme from lattices. In the scheme, user identities are computed as matrices serving as public keys, improving secret key extraction efficiency. Re-encryption keys are generated from user public keys, providing non-interactivity. A public accountability algorithm is used to curb malicious proxy behavior in abusing re-encryption keys. The scheme possesses unidirectionality and multi-hop properties. Security analysis demonstrates that the scheme satisfies IND-aID-CPA security in the standard model, while efficiency analysis shows advantages in storage space and performance. However, there remains room for optimization in computational efficiency, and constructing more efficient lattice-based accountable proxy re-encryption schemes represents a future research direction.

References

[1] Blaze M, Bleumer G, Strauss M, et al. Divertible protocols and atomic proxy cryptography [C]// Theory and Application of Cryptographic Techniques, 1998: 127-144.

[2] Green M, Ateniese G. Identity-Based Proxy Re-encryption [C]// Applied Cryptography and Network Security, 2007: 288-306.

[3] Xagawa K. Cryptography with Lattices [D]. Ph. D. dissertation, Tokyo Institute of Technology, 2010. http://xagawa.net/pdf/2010Thesis.pdf.

[4] Singh K, Rangan C P, Banerjee A K. Lattice-based identity based unidirectional proxy re-encryption scheme [C]// International Conference on Security, Privacy, and Applied Cryptography Engineering, Pune, India, 2014: 76–91.

[5] Tang Yongli, Liu Qi, Zhang Xiaohang, et al. Identity-based proxy re-encryption scheme based on RLWE problem [J]. Application Research of Computers, 2021, 38 (04): 1199-1202.

[6] Kim K S, Jeong I R. Collusion-resistant unidirectional proxy re-encryption scheme from lattices [J]. Journal of Communications and Networks, 2016: 1-7.

[7] Wang X Y, Hu A Q, Hao F. Improved collusion-resistant unidirectional proxy re-encryption scheme from lattice [J]. IET Information Security, 2020: 342-351.

[8] Dutta P, Susilo W, Duong D H, et al. Collusion-Resistant Identity-based Proxy Re-Encryption: Lattice-based Constructions in Standard Model [J]. Theoretical Computer Science, vol. 871, 2021: 16-29.

[9] Wang X A, Ge Y, Yang X. PRE+: Dual of proxy re-encryption and its application [C]// Cryptology ePrint Archive, 2013.

[10] Singh K, Rangan C P, Agrawal R, et al. Provably secure lattice based identity based unidirectional PRE and PRE+ schemes [J]. Journal of Information Security and Applications, 2020, 54 (3/4): 102569.

[11] Ateniese G, Fu K, Green M, et al. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage [J]. ACM Transactions on Information and System Security, vol. 9, no. 1, 2006: 1-30.

[12] Guo H, Zhang Z F, Xu J, et al. Non-Transferable Proxy Re-Encryption [J]. The Computer Journal, vol. 62, no. 4, 2019: 490-506.

[13] Guo H, Zhang Z F, Xu J, et al. Accountable Proxy Re-Encryption for Secure Data Sharing [J]. IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 1, 2021: 145-159.

[14] Ajtai M. Generating hard instances of lattice problems [C]// The 28th ACM Symposium on Theory of Computing. New York: ACM, 1996: 99-108.

[15] Gentry C, Peikert C, Vaikuntanathan V. How to use a short basis: Trapdoors for hard lattices and new cryptographic constructions [C]// The 40th ACM Symposium on Theory of Computing, Victoria, Canada, 2008: 197–206.

[16] Wang Fenghe, Hu Yupu, Jia Yanyan. Lattice-based signature scheme in the standard model [J]. Journal of Xidian University, 2012, 39 (04): 57-61, 119.

[17] Micciancio, D, Peikert C. Hardness of SIS and LWE with Small Parameters [C]// The 33rd Annual International Cryptology Conference, vol. 2013: 21-39.

[18] Regev O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography [C]// Proceedings of the 37th annual ACM symposium on Theory of computing. ACM, 2005: 84-93.

[19] Wang X Y, Hu A Q, Hao F. Feasibility Analysis of Lattice-Based Proxy Re-Encryption [C]// Proc of the 17th International Conference on Cryptography, Security and Privacy, 2017: 12-16.

[20] Hou J, Jiang M, Guo Y, et al. Efficient identity-based multi-bit proxy re-encryption over lattice in the standard model [J]. Information Security Technical Report, 2019: 329-334.

[21] Kirshanova E. Proxy re-Encryption from lattices [C]// Proc of the 17th International Conference on Public-Key Cryptography Volume 8383, 2014: 77-94.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.