Multi-Hop Fine-Grained Proxy Re-Encryption

Yunxiao Zhou[1,2], Shengli Liu[2,3](✉), and Shuai Han[1,2](✉)

[1] School of Cyber Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{cloudzhou,dalen17}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
slliu@sjtu.edu.cn

Abstract. Proxy re-encryption (PRE) allows a proxy to transform a ciphertext intended for Alice (delegator) to another ciphertext intended for Bob (delegatee) without revealing the underlying message. Recently, a new variant of PRE, namely fine-grained PRE (FPRE), was proposed in [Zhou et al., Asiacrypt 2023]. Generally, FPRE is designed for a function family $\mathcal{F}$: each re-encryption key $\mathsf{rk}^f_{A\to B}$ is associated with a function $f \in \mathcal{F}$, and with $\mathsf{rk}^f_{A\to B}$, a proxy can transform Alice's ciphertext encrypting m to Bob's ciphertext encrypting $f(m)$. However, their scheme only supports single-hop re-encryption and achieves only CPA security.

In this paper, we formalize *multi-hop* FPRE (mFPRE) that supports multi-hop re-encryptions in the fine-grained setting, and propose two mFPRE schemes achieving CPA security and stronger HRA security (security against honest re-encryption attacks), respectively.

- For multi-hop FPRE, we formally define its syntax and formalize a set of security notions including CPA security, HRA security, undirectionality and ciphertext unlinkablity. HRA security is stronger and more reasonable than CPA security, and ciphertext unlinkablity blurs the proxy relations among a chain of multi-hop re-encryptions, hence providing better privacy. We establish the relations between these security notions.
- Our mFPRE schemes support fine-grained re-encryptions for bounded linear functions and have security based on the learning-with-errors (LWE) assumption in the standard model. In particular, one of our schemes is HRA secure and enjoys all the aforementioned desirable securities. To achieve CPA security and HRA security for mFPRE, we extend the framework of [Jafargholi et al., Crypto 2017] and the technique of the [Fuchsbauer et al., PKC 2019].

1 Introduction

Proxy re-encryption (PRE) extends the functionality of public-key encryption with re-encryption capability [7]. Let $(pk^{(A)}, sk^{(A)})$ and $(pk^{(B)}, sk^{(B)})$ be Alice and Bob's public and secret keys, respectively. Then Alice can generate a re-encryption key $\mathsf{rk}_{A\to B}$ with her key pair $(pk^{(A)}, sk^{(A)})$ and Bob's public key $pk^{(B)}$, and issue $\mathsf{rk}_{A\to B}$ to a proxy. Later her proxy is able to transform Alice's ciphertext $ct^{(A)}$ encrypting a message m to Bob's ciphertext $ct^{(B)}$ encrypting the same message, but the proxy cannot learn any information about m from $ct^{(A)}$, $ct^{(B)}$ and $\mathsf{rk}_{A\to B}$. Since its introduction, PRE has found a variety of applications, like email forwarding system [7], secure distributed file system [5], digital rights management system [23], etc.

If the re-encryption key $\mathsf{rk}_{A\to B}$ can implement ciphertext transform not only from Alice to Bob, but also vice verse, then the PRE scheme is a *bidirectional* one. In contrast, if $\mathsf{rk}_{A\to B}$ does not support ciphertext transformation from Bob to Alice, then the PRE scheme is a *unidirectional* one. Note that the unidirectional property captures a more precise re-encryption authorization than the bidirectional property. Meanwhile, a unidirectional PRE can support bidirectional re-encryption authorization by issuing both $\mathsf{rk}_{A\to B}$ and $\mathsf{rk}_{B\to A}$ to a proxy. Therefore, unidirectional PRE is more welcome. However, designing unidirectional PREs is more challenging than its bidirectional siblings. In this paper, we focus on unidirectional PRE.

After transformation from $ct^{(A)}$ to $ct^{(B)}$ with $\mathsf{rk}_{A\to B}$, if the resulting $ct^{(B)}$ cannot be further transformed, the PRE scheme is a *single-hop* one. Otherwise, the resulting $ct^{(B)}$ can be further transformed to Charlie's ciphertext $ct^{(C)}$ with $\mathsf{rk}_{B\to C}$ (and so on), then the PRE scheme becomes a *multi-hop* one. Multi-hop PRE schemes support ciphertext transformation chains and provide re-encryption services in a more convenient way.

Fine-Grained Proxy Re-Encryption. Traditionally, PRE provides an all-or-nothing authorization with which either the receiver can decrypt the transformed ciphertext to obtain the whole message m, or it learns nothing about m. Recently, PRE was further extended to support fine-grained re-encryption authorization in [24], and this variant PRE is named *fine-grained* PRE (FPRE). In an FPRE scheme, the re-encryption key $\mathsf{rk}^{f}_{A\to B}$ is further equipped with a function f which captures the precise re-encryption ability granted to a proxy. With $\mathsf{rk}^{f}_{A\to B}$, the proxy can transform Alice's ciphertext $ct^{(A)}$ encrypting a message m to Bob's ciphertext $ct^{(B)}$ encrypting $f(m)$ under $pk^{(B)}$. The recent work in [24] constructed a single-hop unidirectional FPRE scheme w.r.t. bounded linear functions, and proved its CPA security based on the learning-with-errors (LWE) assumption. However, there are two limitations in the FPRE scheme [24].

- The scheme only supports single-hop re-encryption. Suppose that Alice's ciphertext $ct^{(A)}$ has been transformed to a re-encrypted ciphertext $ct^{(B)}$ for Bob. Now Bob wants to forward the underlying message to Charlie, but he can not ask his proxy to do the ciphertext transformation for him due to the single-hop limitation of the FPRE. Thus, he has to decrypt $ct^{(B)}$ to recover the message and encrypt that message under Charlie's public key by himself. The decrypt-then-encrypt operation imposes extra working load to Bob. With a multi-hop FPRE scheme, this job becomes easy. Bob can simply forward the ciphertext $ct^{(B)}$ to his proxy and his proxy will be in charge of the ciphertext transformation.

- The scheme only achieves CPA security. In their CPA model, the adversary is not allowed to learn any re-encryptions from the target user to corrupted users. This is not reasonable. Consider such a scenario: Alice has sent a ciphertext $ct^{(A)}$ to her proxy and her proxy has transformed $ct^{(A)}$ to a re-encrypted ciphertext $ct^{(B)}$ for Bob. Now Bob is corrupted by an adversary. Later, Alice receives a new ciphertext $ct^{*(A)}$, and it is natural to require that the adversary learns nothing about the underlying message of $ct^{*(A)}$. However, this desired security cannot be guaranteed by CPA security since in the CPA model, the adversary is not allowed to learn any re-encryptions from the target user Alice to a corrupted user Bob.

  In fact, obtaining re-encryptions from the target user to a corrupted user is the so-called honest re-encryption attacks (HRA) [9]. When taking HRA attacks into account, the CPA security is lifted to HRA security. As demonstrated in [9], HRA security is more reasonable than CPA security.

The above two limitations lead to an interesting question:

Can we construct a multi-hop fine-grained PRE scheme, preferably also achieving HRA security?

Related Works on Multi-Hop PRE Schemes. There already exist some unidirectional multi-hop PRE schemes in the literature. Chandran et al. [8] designed the first multi-hop unidirectional PRE scheme from program obfuscation and showed the selective obfuscation-based security of their schemes from the LWE assumption. Phong et al. [21] proposed a multi-hop PRE scheme with selective CPA security. However, their scheme is interactive, i.e., the re-encryption key generation algorithm requires both user i and user j's secret keys. Lai et al. [16] proposed a multi-hop PRE scheme achieving selective CCA security from indistinguishability obfuscation (iO). However, iO is a theoretical tool and far from being practical. Fan et al. [11] presented a latticed-based scheme, achieving selective tag-based CCA (tbCCA) security, but proxy relations (i.e., challenge graph of the adversary) are restricted to tree structure. Note that the tbCCA security and the HRA security are not comparable since tbCCA security model does not capture honest re-encryption attacks. Later, Fuchsbauer et al. [12] improved Chandran et al.'s scheme [8] to HRA security based on LWE. At the same time, they presented another multi-hop unidirectional scheme constructed from fully homomorphic encryption [13] and also achieved HRA security from LWE on the ideal lattices and circular-security assumption. Recently, Miao et al. [18] proposed a generic construction of multi-hop PRE with selective HRA security, and presented instantiations based on the decisional Diffie-Hellman (DDH) assumption.

All the existing multi-hop PRE schemes do not consider the fine-grained re-encryption, so the multi-hop *fine-grained* PRE with HRA security is still missing.

Our Contributions. In this work, we propose the first *multi-hop fine-grained* PRE scheme from LWE in the standard model.

- *Formal Definitions for Multi-Hop Fine-Grained PRE and Its Securities.* We formalize multi-hop fine-grained PRE (mFPRE) that supports multiple re-encryptions in the fine-grained setting. We also present the formal CPA and HRA security notions for multi-hop FPRE. In addition, we define unidirectionality (UNID) and ciphertext unlinkability (CUL) for mFPRE. The CUL security guarantees that the chain of multi-hop re-encryptions does not leak information about proxy relations among them, and hence provide better privacy. Moreover, we prove that UNID is implied by CPA, and CUL is implied by CPA and a weak security notion named source-hiding (SH).
- *Generic Framework for Achieving CPA and HRA Security for Multi-Hop FPRE.* We extend the framework in [15] and adapt the techniques in [12] to the multi-hop FPRE setting for achieving (adaptive) CPA and HRA security. More precisely, we first define three weaker security notions including indistinguishability (IND), weak key-privacy (wKP) and source-hiding (SH). Then, we show that the CPA security of multi-hop FPRE is implied by IND and wKP, and the HRA security is implied by IND, wKP and SH. For proxy relations being chains or trees, our reduction only loses a quasi-polynomial factor. Note that the chain and tree topology have good applications in encrypted cloud storage, encrypted email forwarding, etc., as noted by [12].
- *Construction of Multi-Hop FPRE from LWE.* We propose two unidirectional multi-hop FPRE schemes, including a CPA secure mFPRE_1 and an HRA secure mFPRE_2, for bounded linear functions[1]. More precisely, we prove that our first scheme mFPRE_1 has IND and wKP securities and hence achieves CPA security and UNID security, and prove that our second scheme mFPRE_2 has IND, wKP and SH securities and hence achieves HRA security, UNID security and CUL security. Both of the schemes are based on the LWE assumption in the standard model.

We refer to Fig. 1 for an overview of the security notions for multi-hop FPRE and their relations established in this work, and refer to Table 1 for a comparison of our schemes with known multi-hop unidirectional PRE schemes.

Technical Overview. Below we give a high-level overview of our multi-hop fine-grained PRE (mFPRE) scheme. We will first review the single-hop FPRE scheme proposed in [24]. Then we will explain how we realize multi-hop re-encryptions and how we achieve HRA security. For simplicity, we do not specify the dimensions of matrices/vectors.

Recap: The Single-Hop FPRE Scheme in [24] and Its Limitations. We give a brief description of the single-hop scheme in [24]. For user i, its public key $pk^{(i)}$ consists of two matrices $\mathbf{A}_1^{(i)} = \binom{\overline{\mathbf{A}}_1^{(i)}}{\underline{\mathbf{A}}_1^{(i)}}$ and $\mathbf{A}_2^{(i)} = \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}_2^{(i)}}$, and its secret key

[1] Here “bounded” mean that the coefficients are of bounded norm. We note that the existing (single-hop) FPRE schemes [24] are also w.r.t. bounded linear functions.

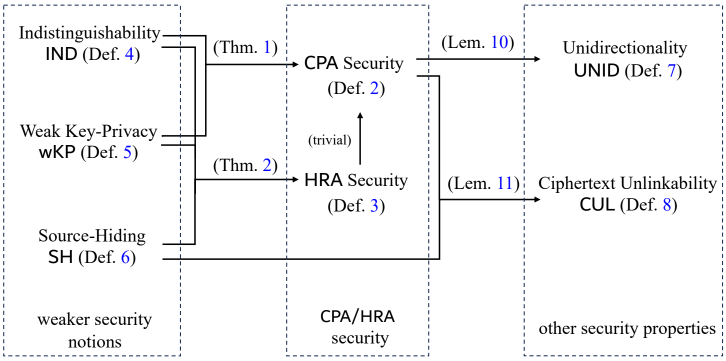

Fig. 1. Security notions of multi-hop FPRE and their relations.

$sk^{(i)}$ contains a trapdoor $\mathbf{T}^{(i)}$ of $\overline{\mathbf{A}}_1^{(i)}$.[2] Here the upper part of $\mathbf{A}_2^{(i)}$ is a (fixed) matrix $\overline{\mathbf{A}}$ generated by a trusted setup and shared by all users, as required by the security of the scheme [24].

The ciphertexts of their scheme have two levels. The first-level/second-level ciphertext $ct_1^{(i)}/ct_2^{(i)}$ of user i is generated using $\mathbf{A}_1^{(i)}/\mathbf{A}_2^{(i)}$ in $pk^{(i)}$ according to the dual Regev encryption scheme [22], namely for level $b \in \{1,2\}$,

$$ct_b^{(i)} = \mathbf{A}_b^{(i)}\mathbf{s} + \mathbf{e} + \begin{pmatrix}\mathbf{0}\\ \lfloor q/2\rfloor \mathbf{m}\end{pmatrix} = \begin{pmatrix}\overline{\mathbf{A}}_b^{(i)}\mathbf{s}+\overline{\mathbf{e}}\\ \underline{\mathbf{A}}_b^{(i)}\mathbf{s}+\underline{\mathbf{e}}+\lfloor q/2\rfloor\cdot\mathbf{m}\end{pmatrix}, \tag{1}$$

where $\mathbf{s}$ and $\mathbf{e} = \binom{\overline{\mathbf{e}}}{\underline{\mathbf{e}}}$ are sampled according to a noise distribution χ.

To realize fine-grained re-encryptions w.r.t. a linear function $f_{\mathbf{M}} : \mathbf{m} \mapsto \mathbf{M}\cdot\mathbf{m}$, the re-encryption key is defined as $\mathsf{rk}_{i\to j}^{f_{\mathbf{M}}} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$, where $\mathbf{R}$ is a small norm matrix satisfying

$$\mathbf{R}\overline{\mathbf{A}}_1^{(i)} = \mathbf{A}_2^{(j)}\mathbf{S}+\mathbf{E}-\binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}_1^{(i)} \tag{2}$$

with matrices $\mathbf{S}, \mathbf{E}$ following the noise distribution χ. Such $\mathbf{R}$ can be efficiently found by using the pre-image sampling algorithm $\mathsf{SamplePre}$ in [14] with the help of the trapdoor $\mathbf{T}^{(i)}$ of $\overline{\mathbf{A}}_1^{(i)}$ contained in $sk^{(i)}$ (cf. Footnote 2). Now with $\mathsf{rk}_{i\to j}^{f_{\mathbf{M}}}$, user i's first-level ciphertext $ct_1^{(i)}$ of $\mathbf{m}$ can be converted to user j's second-level

[2] With the trapdoor $\mathbf{T}^{(i)}$ of $\overline{\mathbf{A}}_1^{(i)}$, one can use the pre-image sampling algorithm $\mathsf{SamplePre}$ developed in [14] to sample a small-norm $\mathbf{R}$ such that $\mathbf{R}\cdot\overline{\mathbf{A}}_1^{(i)} = \mathbf{B}$ holds, given any $\mathbf{B}$. We refer to Lemma 3 for more details.

Table 1. Comparison of multi-hop unidirectional PRE schemes. The column **Standard Model?** asks whether the security is proved in the standard model. The column **Adaptive Corruptions?** asks whether all the security notions support adaptive corruptions. The column **Security** shows the type of security that the scheme achieves, where "HRA" refers to security against honest re-encryption attacks [9], and "tbCCA" refers to tag-based CCA [11] which is incomparable with HRA and restricts the proxy relations (i.e., challenge graph) to tree structure. The column **UNID** shows whether the scheme has unidirectionality. The column **CUL** shows whether the scheme has ciphertext unlinkability. The column **Assumption** shows the assumptions that the security of the scheme is based on, where "iO" refers to indistinguishability obfuscation. The column **Post Quantum?** asks whether the scheme is based on a post-quantum assumption. The column **Fine-Grained?** asks whether the scheme supports fine-grained re-encryptions. The column **Maximum Hops** shows the maximum re-encryption hops that the scheme supports, where "poly-log" refers to $\mathsf{poly}(\log \lambda)$, "sub-linear" refers to λ^{ε} with $0 < \varepsilon < 1$ in the security parameter λ, and "unbounded*" means that the PRE scheme in [18] can support any number of re-encryptions, but at the cost that the ciphertext length grows linearly with the number of re-encryptions. "–" means that no proof or discussion is provided.

PRE Scheme	Standard Model?	Adaptive Corruptions?	Security	UNID	CUL	Assumption	Post Quantum?	Fine–Grained?	Maximum Hops
FL19 [11]	✓	×	tbCCA	✓	–	LWE	✓	–	poly-log
LHAM20 [17]	✓	×	CCA	✓	–	iO	×	–	–
MPW23 [18]	✓	×	HRA	✓	–	DDH	×	–	unbounded*
FKKP19 [12]+ CCLNX14 [8]	✓	✓	HRA	✓	✓	LWE	✓	–	sub-linear
FKKP19 [12] +Gen09 [13]	✓	✓	HRA	✓	✓	LWE over ideal lattice + circular security	✓	–	–
mFPRE_1	✓	✓	CPA	✓	–	LWE	✓	✓	sub-linear
mFPRE_2	✓	✓	HRA	✓	✓	LWE	✓	✓	sub-linear

ciphertext $ct_2^{(j)}$ of the linear function $\mathbf{M} \cdot \mathbf{m}$ via multiplication

$$
\begin{aligned}
ct_2^{(j)} := \mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \cdot ct_1^{(i)} &= \left(\mathbf{R} \;\middle|\; \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right) \cdot \begin{pmatrix} \overline{\mathbf{A}}_1^{(i)}\mathbf{s} + \overline{\mathbf{e}} \\ \underline{\mathbf{A}}_1^{(i)}\mathbf{s} + \underline{\mathbf{e}} + \lfloor q/2 \rfloor \cdot \mathbf{m} \end{pmatrix} \\
&= \underbrace{\left(\mathbf{R}\overline{\mathbf{A}}_1^{(i)} + \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)\underline{\mathbf{A}}_1^{(i)}\right)}_{=\mathbf{A}_2^{(j)}\mathbf{S}+\mathbf{E} \text{ by } (2)} \cdot \mathbf{s} + \mathbf{R}\overline{\mathbf{e}} + \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\underline{\mathbf{e}}\end{smallmatrix}\right) + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2 \rfloor \cdot \mathbf{Mm}\end{smallmatrix}\right) \\
&= \mathbf{A}_2^{(j)} \underbrace{\mathbf{Ss}}_{:=\mathbf{s}'} + \underbrace{\mathbf{Es} + \mathbf{R}\overline{\mathbf{e}} + \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\underline{\mathbf{e}}\end{smallmatrix}\right)}_{:=\mathbf{e}'} + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2 \rfloor \cdot \mathbf{Mm}\end{smallmatrix}\right). \qquad (3)
\end{aligned}
$$

Though a first-level ciphertext $ct_1^{(i)}$ can be re-encrypted to a second-level ciphertext $ct_2^{(j)}$, a second-level ciphertext $ct_2^{(j)}$ cannot be re-encrypted furthermore (no matter to first- or second-level ciphertexts), as explained below.

- To enable further re-encryptions of $ct_2^{(j)}$ to another user (say user k), user j need to compute a re-encryption key $\mathsf{rk}_{j \to k}^{f_{\mathbf{M}'}}$ similar to (2), and in particular, user j need to compute a small-norm $\mathbf{R}$ satisfying

$$\mathbf{R}\overline{\mathbf{A}} = \mathbf{A}_b^{(k)}\mathbf{S} + \mathbf{E} - \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}'\end{smallmatrix}\right)\underline{\mathbf{A}}_2^{(j)} \quad \text{for some } b \in \{1,2\}, \tag{4}$$

where $\overline{\mathbf{A}}$ is the upper part of $\mathbf{A}_2^{(j)}$.

- Note that $\overline{\mathbf{A}}$ is chosen by a trusted setup, so user j has no trapdoor of $\overline{\mathbf{A}}$. This is crucial to the security of their single-hop scheme [24], since their security proof needs to embed an LWE instance to $\overline{\mathbf{A}}$. But without knowing a trapdoor of $\overline{\mathbf{A}}$, user j *cannot* generate a $\mathbf{R}$ satisfying (4). [3]

Overall, it is the security that limits the scheme in [24] serving only for *single-hop* re-encryptions.

Achieving Multi-Hop Re-Encryptions. Note that in the single-hop scheme [24], the ciphertexts $ct_1^{(i)}, ct_2^{(i)}$ of two levels have an almost identical form (i.e., the dual Regev encryption) except for the matrix ($\mathbf{A}_1^{(i)}$ or $\mathbf{A}_2^{(i)}$) used in the encryption. The first-level ciphertext $ct_1^{(i)}$ can be re-encrypted since user i has the trapdoor of $\overline{\mathbf{A}}_1^{(i)}$, while the second-level ciphertext $ct_2^{(i)}$ cannot since user i does not have the trapdoor of $\overline{\mathbf{A}}$.

To enable multi-hop re-encryptions, the public key $pk^{(i)}$ in our scheme contains only one matrix $\mathbf{A}^{(i)} = \left(\begin{smallmatrix}\overline{\mathbf{A}}^{(i)}\\ \underline{\mathbf{A}}^{(i)}\end{smallmatrix}\right)$, and the secret key $sk^{(i)}$ is the trapdoor $\mathbf{T}^{(i)}$ of $\overline{\mathbf{A}}^{(i)}$. (So our scheme has a transparent setup in contrast to [24].) The ciphertexts $ct^{(i)}$ in our scheme stick to $\mathbf{A}^{(i)}$ during encryption, i.e.,

$$ct^{(i)} = \mathbf{A}^{(i)}\mathbf{s} + \mathbf{e} + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2\rfloor\mathbf{m}\end{smallmatrix}\right).$$

The re-encryption key $\mathsf{rk}_{i\to j}^{f_{\mathbf{M}}} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$ in our scheme generates the small norm $\mathbf{R}$ according to

$$\mathbf{R}\overline{\mathbf{A}}^{(i)} = \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)\underline{\mathbf{A}}^{(i)}.$$

In a nutshell, we discard the subscripts $1, 2$ in our scheme.

Similar to the analysis (3), in our scheme, user i's ciphertext $ct^{(i)} = \mathbf{A}^{(i)}\mathbf{s} + \mathbf{e} + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2\rfloor\cdot\mathbf{m}\end{smallmatrix}\right)$ of message $\mathbf{m}$ can be translated to user j's ciphertext with

$$ct^{(j)} := \mathsf{rk}_{i\to j}^{f_{\mathbf{M}}} \cdot ct^{(i)} = \mathbf{A}^{(j)} \underbrace{\mathbf{S}\mathbf{s}}_{:=\mathbf{s}'} + \underbrace{\mathbf{E}\mathbf{s} + \mathbf{R}\overline{\mathbf{e}} + \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\underline{\mathbf{e}}\end{smallmatrix}\right)}_{:=\mathbf{e}'} + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2\rfloor\cdot\underbrace{\mathbf{M}\cdot\mathbf{m}}_{=f_{\mathbf{M}}(\mathbf{m})}\end{smallmatrix}\right). \tag{5}$$

[3] Otherwise, assuming that user j can generate a $\mathbf{R}$ satisfying (4) without knowing a trapdoor of $\overline{\mathbf{A}}$, then anyone (including user k) can generate such $\mathbf{R}$ and thus $\mathsf{rk}_{j\to k}^{f_{\mathbf{M}'}}$ without the help of user j. In this case, user k can translate all ciphertexts $ct_2^{(j)}$ intended for j to ciphertexts $ct_b^{(k)}$ ($b \in \{1,2\}$) encrypted under $pk^{(k)}$ by itself, and then decrypt the re-encrypted ciphertexts using $sk^{(k)}$ to learn information about the message underlying $ct_2^{(j)}$, violating the confidentiality of encryption scheme.

Now in our scheme, user j owns the trapdoor $\mathbf{T}^{(j)}$ of $\overline{\mathbf{A}}^{(j)}$ in its secret key, so it is able to generate $\mathsf{rk}_{j\to k}^{f_{\mathbf{M}'}} := \left(\mathbf{R}' \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}'\end{smallmatrix}\right)$ by sampling a small norm $\mathbf{R}'$ satisfying

$$\mathbf{R}'\overline{\mathbf{A}}^{(j)} = \mathbf{A}^{(k)}\mathbf{S}' + \mathbf{E}' - \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}'\end{smallmatrix}\right)\underline{\mathbf{A}}^{(j)}.$$

Consequently, with $\mathsf{rk}_{j\to k}^{f_{\mathbf{M}'}}$, the re-encryption $ct^{(j)} = \mathbf{A}^{(j)}\mathbf{s}' + \mathbf{e}' + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2\rfloor\cdot\mathbf{M}\cdot\mathbf{m}\end{smallmatrix}\right)$ generated by (5) can be further re-encrypted to user k's ciphertext

$$ct^{(k)} := \mathsf{rk}_{j\to k}^{f_{\mathbf{M}'}} \cdot ct^{(j)} = \mathbf{A}^{(k)} \underbrace{\mathbf{S}'\mathbf{s}'}_{:=\mathbf{s}''} + \underbrace{\mathbf{E}'\mathbf{s}' + \mathbf{R}'\overline{\mathbf{e}'} + \left(\begin{smallmatrix}\mathbf{0}\\ \mathbf{M}'\underline{\mathbf{e}'}\end{smallmatrix}\right)}_{:=\mathbf{e}''} + \left(\begin{smallmatrix}\mathbf{0}\\ \lfloor q/2\rfloor\cdot\underbrace{\mathbf{M}'\cdot(\mathbf{M}\mathbf{m})}_{=f_{\mathbf{M}'}(f_{\mathbf{M}}(\mathbf{m}))}\end{smallmatrix}\right),$$

which encrypts $f_{\mathbf{M}'}(f_{\mathbf{M}}(\mathbf{m})) := \mathbf{M}' \cdot \mathbf{M} \cdot \mathbf{m}$. In this way, the re-encryptions can be further extended with $ct^{(i)} \xrightarrow{\mathsf{rk}_{i\to j}^{f_{\mathbf{M}}}} ct^{(j)} \xrightarrow{\mathsf{rk}_{j\to k}^{f_{\mathbf{M}'}}} ct^{(k)} \xrightarrow{\mathsf{rk}_{k\to w}^{f_{\mathbf{M}''}}} \cdots$, and thus we achieve *multi-hop* fine-grained PRE for linear functions. Note that the norm of the errors $\mathbf{e}, \mathbf{e}', \mathbf{e}'', \cdots$ increases as the re-encryption continues, so to guarantee the correctness of decryption, the re-encryption can go on until the norm of errors reaches $\lfloor q/4 \rfloor$. In fact, our multi-hop FPRE scheme supports constant hops of re-encryptions under polynomial modulus q and supports sub-linear hops of re-encryptions under sub-exponential modulus q.

Overall, since user j has the trapdoor $\mathbf{T}^{(j)}$ of $\overline{\mathbf{A}}^{(j)}$ in our scheme, this rescues our scheme from single-hop, but at the same time, it incurs an issue: we cannot embed an LWE instance to $\overline{\mathbf{A}}^{(j)}$ in the security proof. To avoid this issue, the scheme in [24] prohibits user j from having the trapdoor of both matrices in public key, which in turn limits it to supporting only single-hop re-encryption. To address this issue, we need new techniques to prove security for our mFPRE.

Below we will first show the high-level ideas of the selective CPA security proof of our scheme, and then explain how we upgrade the selective security to adaptive security by adapting the framework of [15, 12] to the fine-grained setting, and explain how we achieve the stronger HRA security.

Selective CPA Security of Our Scheme. We give a high-level overview of the selective CPA security proof of our scheme. Roughly speaking, the (adaptive) CPA security asks the hardness of determining whether a ciphertext ct^* under $pk^{(i^*)}$ encrypts $\mathbf{m}_0$ or $\mathbf{m}_1$, even if an adversary $\mathcal{A}$ can get re-encryption keys $\{\mathsf{rk}_{i\to j}^{f}\}$ and secret keys $\{sk^{(i)}\}$ of some users. To prevent trivial attacks, $\mathcal{A}$ cannot corrupt the target user i^*, and cannot obtain a chain of re-encryption keys from i^* to some corrupted user j. Selective CPA security is weaker as it requires $\mathcal{A}$ to declare the target user i^* and the tuples (i, j) for which $\mathcal{A}$ wants to obtain the corresponding $\{\mathsf{rk}_{i\to j}^{f}\}$ at the beginning of the game.

The main ideas for the selective CPA security proof are: we first change the generations of re-encryption keys $\{\mathsf{rk}_{i\to j}^{f}\}$ so that it does not involve $sk^{(i^*)}$, and then the indistinguishability of ct^* essentially follows from the CPA security of the dual Regev encryption scheme (based on LWE). More precisely,

- **Step 1. Simulating the generation of $\{\mathsf{rk}_{i\to j}^{f}\}$ without knowing $sk^{(i^*)}$.** Let us take an (acyclic) chain of re-encryption keys $\mathsf{rk}_{i^*\to j_1}^{f_1}, \mathsf{rk}_{j_1\to j_2}^{f_2}, \cdots, \mathsf{rk}_{j_{d-1}\to j_d}^{f_{d-1}}$

as example to show how we simulate them in a computationally indistinguishable way without using $sk^{(i^*)}$.

Observe that only the generation of $\mathsf{rk}^{f_1}_{i^* \to j_1}$ involves $sk^{(i^*)}$, where the trapdoor $sk^{(i^*)} = \mathbf{T}^{(i^*)}$ of $\overline{\mathbf{A}}^{(i^*)}$ is used to sample $\mathbf{R}$ satisfying

$$\mathbf{R}\overline{\mathbf{A}}^{(i^*)} = \colorbox{lightgray}{$\mathbf{A}^{(j_1)}\mathbf{S} + \mathbf{E}$} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(i^*)}.$$

Thus we need an indistinguishable way to sample it without trapdoor $\mathbf{T}^{(i^*)}$.

If we can embed an LWE instance to $\colorbox{lightgray}{$\mathbf{A}^{(j_1)}\mathbf{S} + \mathbf{E}$}$ in the above equation, then it can be replaced by a uniform $\colorbox{lightgray}{$\mathbf{U}$}$, and consequently, we have

$$\mathbf{R}\overline{\mathbf{A}}^{(i^*)} = \colorbox{lightgray}{$\mathbf{A}^{(j_1)}\mathbf{S} + \mathbf{E}$} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(i^*)} \overset{c}{\approx} \colorbox{lightgray}{$\mathbf{U}$} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(i^*)} \equiv \colorbox{lightgray}{$\mathbf{U}$}.$$

As a result, we are able to sample $\mathbf{R}$ such that $\mathbf{R}\overline{\mathbf{A}}^{(i^*)} \equiv \colorbox{lightgray}{$\mathbf{U}$}$ by simply choosing it according to a proper discrete Gaussian distribution.[4] However, we cannot embed the LWE instance, since the trapdoor of $\mathbf{A}^{(j_1)}$ is needed to generate $\mathsf{rk}^{f_2}_{j_1 \to j_2}$. This is exactly the issue we mentioned before.

To solve the problem without sacrificing the capability of multi-hop re-encryptions, we simulate the chain of re-encryption keys in reverse order. We will first change the generation of the very last $\mathsf{rk}^{f_{d-1}}_{j_{d-1} \to j_d}$ in the chain as follows. Since $\mathsf{rk}^{f_{d-1}}_{j_{d-1} \to j_d}$ lies in the very end of the chain, we do not need to generate re-encryption key from user j_d to any other users. Moreover, this chain starting from i^* contains only uncorrupted users to avoid trivial attacks. Consequently, the secret key $sk^{(j_d)}$ of user j_d is in fact not needed in the experiment, and now we can embed an LWE instance to $\mathbf{A}^{(j_d)}\mathbf{S} + \mathbf{E}$ such that

$$\mathbf{R}\overline{\mathbf{A}}^{(j_{d-1})} = \colorbox{lightgray}{$\mathbf{A}^{(j_d)}\mathbf{S} + \mathbf{E}$} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(j_{d-1})} \overset{c}{\approx} \colorbox{lightgray}{$\mathbf{U}$} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(j_{d-1})} \equiv \colorbox{lightgray}{$\mathbf{U}$}.$$

Then $\mathbf{R}$ can be simply sampled following the proper discrete Gaussian distribution so that $\mathbf{R}\overline{\mathbf{A}}^{(j_{d-1})} \equiv \colorbox{lightgray}{$\mathbf{U}$}$.

After the changing of $\mathsf{rk}^{f_{d-1}}_{j_{d-1} \to j_d}$, the secret key $sk^{(j_{d-1})}$ of user j_{d-1} is no longer involved, and thus through a similar analysis, we can then embed an LWE instance to $\mathbf{A}^{(j_{d-1})}\mathbf{S}+\mathbf{E}$ so that the $\mathbf{R}$ in the second last $\mathsf{rk}^{f_{d-2}}_{j_{d-2} \to j_{d-1}}$ can be sampled following discrete Gaussian. By changing the re-encryption keys one by one, we can eventually simulate all re-encryption keys in the chain by simply sampling them according to discrete Gaussian, without $sk^{(i^*)}$.

More generally, the re-encryption keys $\{\mathsf{rk}^{f}_{i \to j}\}$ queried by $\mathcal{A}$ might not be a chain. Nevertheless, we can simulate them in a similar way, roughly by processing all the chains simultaneously and for each chain in reverse order.

[4] By [14], if $\mathbf{R}$ follows a proper discrete Gaussian distribution, then $\mathbf{R}\overline{\mathbf{A}}^{(i^*)}$ is statistically close to the uniform distribution $\mathbf{U}$. We refer to Lemma 3 for more details.

- **Step 2. Computationally hiding $\mathbf{m}_0/\mathbf{m}_1$ in $ct^{(i^*)}$.** After Step 1, $sk^{(i^*)}$ is not used at all, and thus for the challenge ciphertext $ct^{(i^*)} = \mathbf{A}^{(i^*)}\mathbf{s}+\mathbf{e} + \begin{pmatrix}\mathbf{0}\\ \lfloor q/2\rfloor \mathbf{m}_\beta\end{pmatrix}$ ($\beta \in \{0,1\}$), we can embed an LWE instance to $\mathbf{A}^{(i^*)}\mathbf{s}+\mathbf{e}$, so that the underlying message $\mathbf{m}_\beta$ is hidden to the adversary $\mathcal{A}$.

Overall, this proof strategy works only in the selective setting, as it requires to know the tuples (i,j) for which $\mathcal{A}$ wants to obtain $\{\mathsf{rk}^f_{i\to j}\}$ in advance, so that they can be properly simulated (i.e., in reverse order for each chain).

To achieve adaptive security, if we guess the tuples (i,j) that $\mathcal{A}$ wants to query at the beginning of game, it will incur a security loss as large as $O(2^{\mathfrak{n}^2})$ with $\mathfrak{n}$ the number of users. To reduce the security loss of adaptive security, we extend the frameworks in [15, 12] to multi-hop FPRE, as explained below.

Achieving Adaptive Security with Jafargholi et al.'s Framework. Jafargholi et al. [15] proposed a generic framework for upgrading selective security to adaptive security with a more fine-grained analysis. Later, Fuchsbauer et al. [12] applied the framework of [15] to the security of (traditional) PRE. In this work, we extend the framework of Jafargholi et al. [15] and the techniques of Fuchsbauer et al. [12] to our multi-hop FPRE.

Roughly speaking, the main observations are: although in the above selective proof strategy, we need the whole information (denoted by w) about the tuples (i,j) that $\mathcal{A}$ wants to query for re-encryption keys, only part of the information (denoted by u) is used in simulating the intermediate hybrids. For example, in the proof strategy shown above, Step 1 consists of many hybrids, while in each hybrid we only change the generation of a single re-encryption key in the chain, so a small amount of information u will be sufficient for the reduction to the LWE assumption; in Step 2, the information of $u := i^*$ is sufficient for the reduction. It is shown in [15] that the security loss in such cases can be limited to the maximum size of the information u used across any two successive hybrids, which might be much smaller than the size of w.

To apply their techniques [15, 12], we abstract two useful yet weaker security notions for our multi-hop FPRE, including indistinguishability (IND) and weak key-privacy (wKP), and then establish a theorem by reducing the adaptive CPA security to IND and wKP with a smaller security loss. Concretely, the two weaker notions exactly correspond to Step 1 and Step 2 in the above proof strategy.

Weak Key-Privacy (wKP). It stipulates that the re-encryption key $\mathsf{rk}^f_{i\to j}$ honestly generated by $sk^{(i)}$ can be indistinguishably changed to a simulated one generated without $sk^{(i)}$ in the view of adversary who gets no secret keys $sk^{(i)}$.

Indistinguishability (IND). It requires the indistinguishability of ciphertext for adversary who gets no re-encryption keys $\mathsf{rk}^f_{i\to j}$ and no secret keys $sk^{(i)}$.

The theorem showing adaptive CPA security based on IND and wKP for our multi-hop FPRE is proved in a similar way as [12, 15]. For an arbitrary adversary who can obtain re-encryption keys $\{\mathsf{rk}^f_{i\to j}\}$ for arbitrary tuples (i,j), the security loss of adaptive CPA security is $\mathfrak{n}^{O(\mathfrak{n})}$ in contrast to the naive guessing strategy

$O(2^{\mathfrak{n}^2})$. In many realistic scenarios like key rotation for encrypted cloud storage or forwarding of encrypted mail, as demonstrated in [12], the proxy relations are in fact *trees, chains or low-depth graphs*. In these situations, an adversary can only obtain $\{\mathsf{rk}_{i \to j}^{f}\}$ for tuples (i, j) that form trees, chains or low-depth graphs, and the security loss is only quasi-polynomial $\mathfrak{n}^{O(\log \mathfrak{n})}$.

Achieving HRA Security. Security against honest re-encryption attacks (HRA) was first introduced by Cohen [9] and is a security notion stronger and more reasonable than CPA. Compared with CPA security, HRA also allows the adversary $\mathcal{A}$ to obtain re-encryptions of ciphertexts from the target user i^* to *corrupted* users, as long as the ciphertexts to be re-encrypted are honestly generated and are not (re-encryptions of) the challenge ciphertext ct^*. Note that HRA security is stronger than CPA: in the CPA experiment, $\mathcal{A}$ cannot obtain a chain of re-encryption keys from i^* to corrupted users in order to prevent trivial attacks, and thus cannot generate re-encryptions from i^* to corrupted users by itself.

In order to achieve HRA security, we need to enhance our aforementioned CPA proof strategy with a new computationally indistinguishable method for simulating the generation of re-encryptions of ciphertexts from the target user i^* to corrupted users without using $sk^{(i^*)}$. Note that the re-encryptions from i^* to corrupted users might be a chain $ct^{(i^*)} \to ct^{(j_1)} \to ct^{(j_2)} \to \cdots \to ct^{(j_d)}$, the generation of which involves a chain of re-encryption keys $\mathsf{rk}_{i^* \to j_1}^{f_1}, \mathsf{rk}_{j_1 \to j_2}^{f_2}, \cdots, \mathsf{rk}_{j_{d-1} \to j_d}^{f_{d-1}}$. However, we cannot use similar techniques as the CPA security proof strategy to replace this chain of re-encryption keys with simulated ones, since the involved users $j_1, j_2, \cdots, j_d$ might be corrupted by $\mathcal{A}$.

To bypass this problem, we will simulate the generation of the chain of re-encryptions $ct^{(i^*)} \to ct^{(j_1)} \to ct^{(j_2)} \to \cdots \to ct^{(j_d)}$ directly, without using any of the re-encryption keys $\mathsf{rk}_{i^* \to j_1}^{f_1}, \mathsf{rk}_{j_1 \to j_2}^{f_2}, \cdots, \mathsf{rk}_{j_{d-1} \to j_d}^{f_{d-1}}$, thus also without using $sk^{(i^*)}$. To this end, we abstract a (weak) security notion called source-hiding (SH) for multi-hop FPRE, by adapting the techniques in [15, 12].

Source-Hiding (SH). It stipulates that the honestly generated re-encryption $ct^{(i)} \to ct^{(j)}$ by using $\mathsf{rk}_{i \to j}^{f}$ can be indistinguishably changed to a simulated one generated without $\mathsf{rk}_{i \to j}^{f}$.

The SH security is exactly what we need to upgrade our CPA security proof strategy to HRA security: roughly speaking, by the SH security, we can change all re-encryptions $ct^{(i)} \to ct^{(j)}$ queried by $\mathcal{A}$ to simulated ones without using re-encryption keys (thus $sk^{(i^*)}$ is not involved); then by the wKP security, we can change all re-encryption keys $\{\mathsf{rk}_{i \to j}^{f}\}$ queried by $\mathcal{A}$ to simulated ones without using $sk^{(i^*)}$; finally, by the IND security, the challenge ciphertext ct^* of the target user i^* hides the underlying message.

For achieving *adaptive* HRA security for multi-hop FPRE, we also extend the framework of Jafargholi et al. [15] and the techniques of Fuchsbauer et al. [12], and establish a theorem by reducing the adaptive HRA security to IND, wKP and SH, with similar security loss.

Finally, we give a high-level overview of our second multi-hop FPRE scheme which additionally satisfies SH security. More precisely, we augment each ciphertext with a level $v \in \mathbb{N}$, and use different noise distribution χ_v for the generation of ciphertexts of different levels. Namely, the v-level ciphertext of user i is now generated by

$$ct_v^{(i)} := \mathbf{A}^{(i)}\mathbf{s} + \mathbf{e} + \binom{\mathbf{0}}{\lfloor q/2 \rceil \mathbf{m}} \qquad \text{with } \mathbf{s} \text{ and } \mathbf{e} \text{ following } \chi_v. \tag{6}$$

Moreover, we randomize the generation of re-encryption $ct_v^{(i)} \rightarrow ct_{v+1}^{(j)}$ with $\mathsf{rk}_{i\rightarrow j}^{f_{\mathbf{M}}}$ by adding noises, i.e., choosing $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{e}}$ according to χ_{v+1} and computing

$$\begin{aligned} ct_{v+1}^{(j)} &:= \mathsf{rk}_{i\rightarrow j}^{f_{\mathbf{M}}} \cdot ct_v^{(i)} + \boxed{\mathbf{A}^{(j)}\tilde{\mathbf{s}} + \tilde{\mathbf{e}}} \\ &= \mathbf{A}^{(j)} \underbrace{\mathbf{S}\mathbf{s}}_{:=\mathbf{s}'} + \underbrace{\mathbf{E}\mathbf{s} + \mathbf{R}\bar{\mathbf{e}} + \binom{\mathbf{0}}{\mathbf{M}\underline{\mathbf{e}}}}_{:=\mathbf{e}'} + \binom{\mathbf{0}}{\lfloor q/2 \rceil \cdot \underbrace{\mathbf{M} \cdot \mathbf{m}}_{=f_{\mathbf{M}}(\mathbf{m})}} + \boxed{\mathbf{A}^{(j)}\tilde{\mathbf{s}} + \tilde{\mathbf{e}}} && (7) \\ &= \mathbf{A}^{(j)}(\boxed{\tilde{\mathbf{s}}} + \mathbf{s}') + (\boxed{\tilde{\mathbf{e}}} + \mathbf{e}') + \binom{\mathbf{0}}{\lfloor q/2 \rceil \cdot \mathbf{M} \cdot \mathbf{m}}, && (8) \end{aligned}$$

where (7) follows from (5). By choosing the noise distribution χ_v carefully, we can ensure that $\boxed{\tilde{\mathbf{s}}}$ smudges $\mathbf{s}'$ and $\boxed{\tilde{\mathbf{e}}}$ smudges $\mathbf{e}'$. Consequently, the honestly generated re-encryption $ct_{v+1}^{(j)}$ in (8) is statistically indistinguishable from a freshly generated $(v+1)$-level ciphertext of user j that encrypts $\mathbf{M} \cdot \mathbf{m}$ according to (6), without using $\mathsf{rk}_{i\rightarrow j}^{f_{\mathbf{M}}}$. This shows the SH security of this scheme. Similar to our first scheme, this scheme also achieves IND and wKP securities, thus achieving adaptive HRA security via the generic theorem.

Interestingly, we also show that the SH security together with the CPA security (or HRA security) imply *ciphertext unlinkability* (CUL), which can blur the proxy relations in a chain of multi-hop re-encryptions in a more complex setting.

Relations to Existing Works. Finally, we summarize the results already known in the non-fine-grained setting or in the single-hop fine-grained setting, and the results that are novel in our work.

The weaker security notions $\mathsf{IND}, \mathsf{wKP}, \mathsf{SH}$ were originally defined by Fuchsbauer et al. [12] for (non-fine-grained) PRE. Fuchsbauer et al. [12] also established two theorems showing adaptive CPA security based on IND and wKP and showing adaptive HRA security based on IND, wKP and SH, respectively, for (non-fine-grained) PRE, building upon the framework of Jafargholi et al. [15].

The notion of single-hop FPRE and its CUL security were recently introduced by Zhou et al. [24], where they also formally proved the relation that CPA implies UNID for single-hop FPRE.

In our work, we propose the concept of multi-hop FPRE to support multi-hop fine-grained re-encryptions, and formalize a set of security notions $\mathsf{CPA}, \mathsf{HRA}, \mathsf{IND}, \mathsf{wKP}, \mathsf{SH}, \mathsf{UNID}, \mathsf{CUL}$ in the multi-hop fine-grained setting. Moreover, we establish several useful relations between these security notions for multi-hop FPRE, by adapting the two theorems in [12] and the relation in [24] to our multi-hop FPRE. Besides, we show the relation that $\mathsf{SH} + \mathsf{CPA} \Rightarrow \mathsf{CUL}$ holds for our multi-hop FPRE, which is for the first time established for PRE (no matter in which setting). Furthermore, we construct two multi-hop FPRE schemes from LWE, and prove their $\mathsf{IND}, \mathsf{wKP}$ and SH securities based on the LWE assumption in the standard model, which are novel in our work. According to the relations we established (i.e., Theorem 1 and Theorem 2), the two multi-hop FPRE schemes achieves adaptive CPA and adaptive HRA securities, respectively.

2 Preliminaries

Notations. Let $\lambda \in \mathbb{N}$ denote the security parameter throughout the paper, and all algorithms, distributions, functions and adversaries take 1^λ as an implicit input. If x is defined by y or the value of y is assigned to x, we write $x := y$. For $i, j \in \mathbb{N}$ with $i < j$, define $[i, j] := \{i, i+1, ..., j\}$ and $[j] := \{1, 2, ..., j\}$. For a set $\mathcal{X}$, denote by $x \leftarrow_\$ \mathcal{X}$ the procedure of sampling x from $\mathcal{X}$ uniformly at random. If $\mathcal{D}$ is distribution, $x \leftarrow_\$ \mathcal{D}$ means that x is sampled according to $\mathcal{D}$. All our algorithms are probabilistic unless stated otherwise. We use $y \leftarrow_\$ \mathcal{A}(x)$ to define the random variable y obtained by executing algorithm $\mathcal{A}$ on input x. If $\mathcal{A}$ is deterministic we write $y \leftarrow \mathcal{A}(x)$. "PPT" abbreviates probabilistic polynomial-time. Denote by negl some negligible function. By $\Pr_i[\cdot]$ we denote the probability of a particular event occurring in game G_i.

For random variables X and Y, the min-entropy of X is defined as $\mathbf{H}_\infty(X) := -\log(\max_x \Pr[X = x])$, and the statistical distance between X and Y is defined as $\Delta(X, Y) := \frac{1}{2} \cdot \sum_x |\Pr[X = x] - \Pr[Y = x]|$. If $\Delta(X, Y) = \mathsf{negl}(\lambda)$, we say that X and Y are statistically indistinguishable (close), and denote it by $X \approx_s Y$.

Let $n, m, m', q \in \mathbb{N}$, and let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{v} \in \mathbb{Z}_q^n$, $\mathbf{B} \in \mathbb{Z}_q^{m' \times n}$. Define the lattice $\Lambda(\mathbf{A}) := \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{Z}^n\}$, the q-ary lattice $\Lambda_q(\mathbf{A}) := \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m$, its "orthogonal" lattice $\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{x}^\top \mathbf{A} = \mathbf{0} \mod q\}$, and the "shifted" lattice $\Lambda_q^{\mathbf{v}}(\mathbf{A}) := \{\mathbf{r} \in \mathbb{Z}^m \mid \mathbf{r}^\top \mathbf{A} = \mathbf{v}^\top \mod q\}$, which can be further extended to $\Lambda_q^{\mathbf{B}}(\mathbf{A}) := \{\mathbf{R} \in \mathbb{Z}^{m' \times m} \mid \mathbf{RA} = \mathbf{B} \mod q\}$. Let $\|\mathbf{v}\|$ (resp., $\|\mathbf{v}\|_\infty$) denote its ℓ_2 (resp., infinity) norm. For a matrix $\mathbf{A}$, we define $\|\mathbf{A}\|$ (resp., $\|\mathbf{A}\|_\infty$) as the largest ℓ_2 (resp., infinity) norm of $\mathbf{A}$'s rows. A distribution χ is B-bounded if its support is limited to $[-B, B]$. Let $\mathbb{Z}_q$ be the ring of integers modulo q, and its elements are represented by the integers in $(-q/2, q/2]$.

In Appendix A.2, we present necessary lattice backgrounds, including the definitions of discrete Gaussian distribution, LWE assumption, and the $\mathsf{TrapGen}$, Invert, $\mathsf{SamplePre}$ algorithms introduced in [1, 14, 20].

3 Multi-Hop Fine-Grained PRE

In this section, we formalize a new primitive called *Multi-Hop Fine-Grained PRE* (mFPRE), by extending the concept of single-hop FPRE proposed in [24] to support multi-hop of re-encryptions. Compared with (traditional) PRE, FPRE allows fine-grained delegations, by associating re-encryption key $\mathsf{rk}_{i \to j}^f$ with a function f to support the conversion of user i's ciphertext $ct^{(i)}$ encrypting message m to user j's ciphertext $ct^{(j)}$ encrypting the function value $f(m)$. Moreover, in contrast to single-hop FPRE, our multi-hop FPRE supports multiple re-encryptions, namely, user j's re-encrypted ciphertext $ct^{(j)}$ encrypting $f(m)$ can be further re-encrypted to user k's ciphertext $ct^{(k)}$ encrypting $f'(f(m))$ with the help of another $\mathsf{rk}^{f'}_{j\to k}$, and as forth. These multiple re-encryptions can be correctly decrypted to the corresponding function values, as long as the number of re-encryption hops does not exceed the maximum level.

As for security, we formalize the CPA and HRA security for multi-hop FPRE. To achieve both security, we adapt the framework proposed in [15, 12] to fine-grained setting and establish two theorems reducing CPA and HRA to a set of weaker security notions, including indistinguishablity (IND), weak key-privacy (wKP) and source-hiding (SH), for multi-hop FPRE. Furthermore, we introduce some other security properties including unidirectionality (UNID) and ciphertext unlinkability (CUL) for multi-hop FPRE. See Fig. 1 in introduction for an overview of the relations between these security notions.

More precisely, in Subsect. 3.1, we present the syntax of multi-hop FPRE and define its CPA security and HRA security. In Subsect. 3.2, we give the formal definitions of the set of weaker security notions $\mathsf{IND}, \mathsf{wKP}$ and SH, along with two theorems reducing CPA and HRA security to these weaker security notions. Finally in Subsect. 3.3, we define the UNID and CUL security under adaptive corruptions and demonstrate their relations with other security notions.

3.1 Syntax of Multi-Hop FPRE and Its CPA and HRA Security

Definition 1 (Multi-Hop Fine-Grained PRE). *Let $\mathcal{F}$ be a family of functions from $\mathcal{M}$ to $\mathcal{M}$, where $\mathcal{M}$ is a message space. A multi-hop fine-grained proxy re-encryption (multi-hop FPRE) scheme for function family $\mathcal{F}$ is associated with a maximum level $L \in \mathbb{N}$ and defined with a tuple of PPT algorithms* $\mathsf{mFPRE} = (\mathsf{KGen}, \mathsf{FReKGen}, \mathsf{Enc}, \mathsf{FReEnc}, \mathsf{Dec})$.

- $(pk, sk) \leftarrow_{\$} \mathsf{KGen}$*: The key generation algorithm outputs a pair of public key and secret key (pk, sk).*
- $\mathsf{rk}^{f}_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$*: Taking as input a public-secret key pair $(pk^{(i)}, sk^{(i)})$, another public key $pk^{(j)}$ and a function $f \in \mathcal{F}$, the fine-grained re-encryption key generation algorithm outputs a fine-grained re-encryption key $\mathsf{rk}^{f}_{i\to j}$ that allows re-encrypting ciphertexts intended to i into ciphertexts encrypted for j.*
- $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, m, v)$*: Taking as input pk, a message $m \in \mathcal{M}$ and a level $v \in [0, L]$, the encryption algorithm outputs a v-level ciphertext ct_v.*
- $ct^{(j)}_{v+1} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f}_{i\to j}, ct^{(i)}_v, v)$*: Taking as input a re-encryption key $\mathsf{rk}^{f}_{i\to j}$ and a ciphertext $ct^{(i)}_v$ intended for i and its level $v \in [0, L-1]$, the fine-grained re-encryption algorithm outputs a $(v+1)$-level ciphertext $ct^{(j)}_{v+1}$ re-encrypted for j. We denote it by $ct^{(i)}_v \xrightarrow{\mathsf{rk}^{f}_{i\to j}} ct^{(j)}_{v+1}$.*
- $m \leftarrow \mathsf{Dec}(sk, ct)$*: Taking as input a secret key sk and a ciphertext ct, the deterministic decryption algorithm outputs a message m.*

Correctness. *For all* $m \in \mathcal{M}, v \in [0, L]$, $(pk, sk) \leftarrow_{\$} \mathsf{KGen}$, $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, m, v)$, *it holds that* $\mathsf{Dec}(sk, ct_v) = m$.

Fine-Grained L***-Hop Correctness.*** *For all* $m \in \mathcal{M}$, *user indices* $i_0, i_1, \cdots, i_L$, *functions* $f_1, \cdots, f_L \in \mathcal{F}$, $(pk^{(i_j)}, sk^{(i_j)}) \leftarrow_{\$} \mathsf{KGen}$ *with* $j \in [0, L]$, 0*-level ciphertext* $ct_0^{(i_0)} \leftarrow_{\$} \mathsf{Enc}(pk^{(i_0)}, m, 0)$ *and re-encryption hops* $ct_0^{(i_0)} \xrightarrow{\mathsf{rk}^{f_1}_{i_0 \to i_1}} ct_1^{(i_1)} \xrightarrow{\mathsf{rk}^{f_2}_{i_1 \to i_2}} \cdots \xrightarrow{\mathsf{rk}^{f_L}_{i_{L-1} \to i_L}} ct_L^{(i_L)}$, *where each* $\mathsf{rk}^{f_j}_{i_{j-1} \to i_j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i_{j-1})}, sk^{(i_{j-1})}, pk^{(i_j)}, f_j)$ *and each* $ct_j^{(i_j)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f_j}_{i_{j-1} \to i_j}, ct_{j-1}^{(i_{j-1})}, j-1)$, *it holds that for all* $j \in [L]$,

$$\mathsf{Dec}(sk^{(i_j)}, ct_j^{(i_j)}) = f_j(f_{j-1}(\ldots f_1(m))).$$

CPA Security. Below we formalize the indistinguishability of ciphertexts under chosen-plaintext attacks (CPA) for multi-hop FPRE.

Definition 2 (CPA **Security for Multi-Hop FPRE).** *A multi-hop FPRE scheme* mFPRE *is* CPA *secure, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $\mathfrak{n}$, *it holds that* $\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}} \Rightarrow 1] - \frac{1}{2} \right| \le \mathsf{negl}(\lambda)$, *where the experiment* $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ *is defined in Fig. 2.*

```
$\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$:
For $i \in [\mathfrak{n}]$: $(pk^{(i)}, sk^{(i)}) \leftarrow_{\$} \mathsf{KGen}$
$\mathcal{Q}_{rk} := \emptyset$                //record re-encryption key queries
$\mathcal{Q}_c := \emptyset$                 //record corruption queries
$i^* := \perp$                   //record challenge user
$(i^*, m_0, m_1, v, st) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot), \mathcal{O}_{\text{Cor}}(\cdot)}(\{pk^{(i)}\}_{i \in [\mathfrak{n}]})$
If $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$:
    Return $b \leftarrow_{\$} \{0, 1\}$          //avoid TA1, TA2
$\beta \leftarrow_{\$} \{0, 1\}$
$ct_v^* \leftarrow_{\$} \mathsf{Enc}(pk^{(i^*)}, m_\beta, v)$
$\beta' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot), \mathcal{O}_{\text{Cor}}(\cdot)}(st, ct_v^*)$
If $\beta' = \beta$: Return 1;   Else: Return 0

$\mathcal{O}_{\text{ReKey}}(i, j, f)$: //re-encryption key queries
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk} \cup \{(i, j)\}, \mathcal{Q}_c) = 1$:
        Return $\perp$                //avoid TA2
    $\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(i, j)\}$
    $\mathsf{rk}^f_{i \to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    Return $\mathsf{rk}^f_{i \to j}$

$\mathcal{O}_{\text{Cor}}(i)$:               //corruption queries
    If $i = i^*$: Return $\perp$          //avoid TA1
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$:
        Return $\perp$                //avoid TA2
    $\mathcal{Q}_c := \mathcal{Q}_c \cup \{i\}$
    Return $sk^{(i)}$

$\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c)$:          //check TA2
    If $\exists\ (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$
        s.t. $j_t \in \mathcal{Q}_c$ for some $t \ge 1$:
        Return 1
    Else: Return 0
```

Fig. 2. The CPA security experiment $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE. Here CheckTA is a sub-procedure used to check the trivial attacks.

Remark 1 (On the formalization of CPA *security and discussion on trivial attacks).* We formalize the CPA security by defining the experiment $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ in Fig. 2. More precisely, we consider a multi-user setting, and the adversary $\mathcal{A}$ is allowed to make two kinds of oracle queries *adaptively*:

- through $\mathcal{O}_{\text{ReKey}}(i, j, f)$ query, $\mathcal{A}$ can get re-encryption keys $\mathsf{rk}^f_{i \to j}$, and

– through $\mathcal{O}_{\textsc{Cor}}(i)$ query, $\mathcal{A}$ can corrupt user i and obtain its secret key $sk^{(i)}$.

At some point, $\mathcal{A}$ outputs a challenge user i^*, a pair of messages (m_0, m_1) as well as a level v, and receives a challenge ciphertext ct_v^* which encrypts m_β under $pk^{(i^*)}$ at level v, where β is the challenge bit that $\mathcal{A}$ aims to guess.

To prevent trivial attacks from $\mathcal{A}$, we keep track of two sets: $\mathcal{Q}_c$ records the corrupted users, and $\mathcal{Q}_{rk}$ records the tuples (i, j) that $\mathcal{A}$ obtains a re-encryption key $\mathsf{rk}_{i \to j}^{f}$. Based on that, there are two kinds of trivial attacks **TA1-TA2** to obtain information about the plaintext underlying the challenge ciphertext ct_v^*.

TA1: $i^* \in \mathcal{Q}_c$, i.e., $\mathcal{A}$ corrupts user i^* and obtains its secret key $sk^{(i^*)}$. In this case, $\mathcal{A}$ can decrypt ct_v^* directly via $\mathsf{Dec}(sk^{(i^*)}, ct_v^*)$ and recover m_β.

TA2: $\exists (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$ s.t. $j_t \in \mathcal{Q}_c$ for some $t \geq 1$, i.e., $\mathcal{A}$ gets a chain of re-encryption keys $\mathsf{rk}_{i^* \to j_1}^{f_1}, \mathsf{rk}_{j_1 \to j_2}^{f_2}, \ldots, \mathsf{rk}_{j_{t-1} \to j_t}^{f_t}$ starting from the challenge user i^* and ending at some corrupted user j_t for whom $\mathcal{A}$ ever obtains its secret key $sk^{(j_t)}$. In this case, $\mathcal{A}$ can re-encrypt ct_v^* via $ct_v^* \xrightarrow{\mathsf{rk}_{i^* \to j_1}^{f_1}} ct_{v+1}^{(j_1)} \xrightarrow{\mathsf{rk}_{j_1 \to j_2}^{f_2}} \cdots \xrightarrow{\mathsf{rk}_{j_{t-1} \to j_t}^{f_t}} ct_{v+t}^{(j_t)}$, then simply decrypt $ct_{v+t}^{(j_t)}$ with $sk^{(j_t)}$ to obtain a function of m_β. This kind of trivial attacks is checked by the algorithm $\mathsf{CheckTA}$ defined in Fig. 2 throughout the experiment.

As such, we exclude the above trivial attacks in the CPA experiment.

We note that in contrast to the CPA security for PRE defined in [12], our CPA security does not provide a re-encryption oracle for re-encrypting ciphertexts from the challenge user i^* to uncorrupted users $j \notin \mathcal{Q}_c$. This is because in our CPA experiment, $\mathcal{A}$ can obtain re-encryption keys from i^* to $j \notin \mathcal{Q}_c$ through the $\mathcal{O}_{\textsc{ReKey}}$ oracle and do re-encryption itself for such ciphertexts.

HRA Security. Next we formalize the indistinguishability of ciphertexts under honest-re-encryption attacks (HRA) for multi-hop FPRE. Originally, HRA was first introduced by Cohen [9] as a stronger and more reasonable security notion than CPA for PRE. Below we adapt HRA security to the fine-grained setting for mFPRE. Compared with the CPA security, HRA also allows the adversary to have access to a re-encryption oracle $\mathcal{O}_{\textsc{ReEnc}}$, through which the adversary can learn re-encryptions of ciphertexts from the challenge user i^* to *corrupted* users $j \in \mathcal{Q}_c$, as long as the queried ciphertexts are honestly generated and different from (all derivatives of) the challenge ciphertext ct_v^*.

Definition 3 (HRA Security for Multi-Hop FPRE). *A multi-hop FPRE scheme* mFPRE *is* HRA *secure, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $\mathfrak{n}$, *it holds that* $\mathsf{Adv}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}}(\lambda) := \left| \Pr[\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}} \Rightarrow 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$, *where the experiment* $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}}$ *is defined in Fig. 3.*

Remark 2 (On the formalization of HRA *security and discussion on trivial attacks).* We formalize the HRA security by defining the experiment $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}}$ in Fig. 3. More precisely, we consider a multi-user setting, and the adversary $\mathcal{A}$ is allowed to make four kinds of oracle queries *adaptively*:

```
$\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$:
For $i \in [\mathfrak{n}]$:  $(pk^{(i)}, sk^{(i)}) \leftarrow_{\$} \mathsf{KGen}$
$\mathcal{Q}_{rk} := \emptyset$                         //record re-encryption key queries
$\mathcal{Q}_c := \emptyset$                                   //record corruption queries
$i^* := \perp$                                          //record challenge user
$\mathcal{L} := \perp$                          //record honestly generated ciphertexts
$\mathcal{L}^* := \perp$            //record derivatives of the challenge ciphertext
$\mathsf{ctr} := 0$                        //index of honestly generated ciphertexts
$(i^*, m_0, m_1, v, st) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\mathrm{Cor}}(\cdot),\mathcal{O}_{\mathrm{Enc}}(\cdot,\cdot,\cdot),\mathcal{O}_{\mathrm{ReEnc}}(\cdot,\cdot,\cdot,\cdot)}$
                 $(\{pk^{(i)}\}_{i\in[\mathfrak{n}]})$
If $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$:
    Return $b \leftarrow_{\$} \{0,1\}$                            //avoid TA1, TA2
$\beta \leftarrow_{\$} \{0,1\}$
$\mathsf{ctr} := \mathsf{ctr} + 1$
$ct^*_v \leftarrow_{\$} \mathsf{Enc}(pk^{(i^*)}, m_\beta, v)$
$\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, i^*, (ct^*_v, v))\}$
$\mathcal{L}^* := \mathcal{L}^* \cup \{(\mathsf{ctr}, i^*)\}$            //index of challenge ciphertext
$\beta' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\mathrm{Cor}}(\cdot),\mathcal{O}_{\mathrm{Enc}}(\cdot,\cdot,\cdot),\mathcal{O}_{\mathrm{ReEnc}}(\cdot,\cdot,\cdot,\cdot)}(st, ct^*_v)$
If $\beta' = \beta$: Return 1;   Else: Return 0
```

```
$\mathcal{O}_{\mathrm{ReKey}}(i, j, f)$:                       //re-encryption key queries
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$:
        Return $\perp$                                          //avoid TA2
    $\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(i,j)\}$
    $\mathsf{rk}^f_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    Return $\mathsf{rk}^f_{i\to j}$
```

```
$\mathcal{O}_{\mathrm{Cor}}(i)$:                              //corruption queries
    If $\exists(\cdot, i) \in \mathcal{L}^*$:  Return $\perp$ //avoid TA1, TA3
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$:
        Return $\perp$                                 //avoid TA2
    $\mathcal{Q}_c := \mathcal{Q}_c \cup \{i\}$
    Return $sk^{(i)}$

$\mathcal{O}_{\mathrm{Enc}}(i, m, v)$:          //honest encryption queries
    $\mathsf{ctr} := \mathsf{ctr} + 1$
    $ct^{(i)}_v \leftarrow_{\$} \mathsf{Enc}(pk^{(i)}, m, v)$
    $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, i, (ct^{(i)}_v, v))\}$
    Return $(\mathsf{ctr}, ct^{(i)}_v)$

$\mathcal{O}_{\mathrm{ReEnc}}(i, j, f, k)$: //honest re-encryption queries
    If $(k, i) \in \mathcal{L}^*$ and $j \in \mathcal{Q}_c$:
        Return $\perp$                                 //avoid TA3
    Retrieve $(k, i, (ct', v'))$ from $\mathcal{L}$:
        If fails, return $\perp$
    $\mathsf{rk}^f_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    $ct^{(j)}_{v'+1} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^f_{i\to j}, ct', v')$
    $\mathsf{ctr} := \mathsf{ctr} + 1$
    $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, j, (ct^{(j)}_{v'+1}, v'+1))\}$
    If $(k, i) \in \mathcal{L}^*$:   $\mathcal{L}^* := \mathcal{L}^* \cup \{(\mathsf{ctr}, j)\}$
    Return $(\mathsf{ctr}, ct^{(j)}_{v'+1})$

$\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c)$:                      //check TA2
    If $\exists\ (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$
        s.t. $j_t \in \mathcal{Q}_c$ for some $t \geq 1$:
        Return 1
    Else: Return 0
```

Fig. 3. The HRA security experiment $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE. Here the oracles $\mathcal{O}_{\mathrm{ReKey}}$, $\mathcal{O}_{\mathrm{Cor}}$ and the sub-procedure $\mathsf{CheckTA}$ are the same as those in Fig. 2.

- through $\mathcal{O}_{\mathrm{ReKey}}(i, j, f)$ query, $\mathcal{A}$ can get re-encryption keys $\mathsf{rk}^f_{i\to j}$;
- through $\mathcal{O}_{\mathrm{Cor}}(i)$ query, $\mathcal{A}$ can corrupt user i and obtain its secret key $sk^{(i)}$;
- through $\mathcal{O}_{\mathrm{Enc}}(i, m, v)$ query, $\mathcal{A}$ can obtain honestly generated ciphertexts, which are indexed by counters ctr and can be further re-encrypted through $\mathcal{O}_{\mathrm{ReEnc}}$ query;
- through $\mathcal{O}_{\mathrm{ReEnc}}(i, j, f, k)$ query, $\mathcal{A}$ can obtain re-encryptions of honestly generated ciphertexts (including the challenge ciphertext ct^*_v to be defined later, as well as the re-encrypted ciphertexts output by $\mathcal{O}_{\mathrm{ReEnc}}$ previously), where k is the index of the honestly generated ciphertext to be re-encrypted and i, j, f specify the re-encryption key $\mathsf{rk}^f_{i\to j}$ to be used.

At some point, $\mathcal{A}$ outputs a challenge user i^*, a pair of messages (m_0, m_1) as well as a level v, and receives a challenge ciphertext ct^*_v which encrypts m_β under $pk^{(i^*)}$ at level v, where β is the challenge bit that $\mathcal{A}$ aims to guess.

Similar to the CPA security, we also exclude the two trivial attacks **TA1**-**TA2** as defined in Remark 1, from which $\mathcal{A}$ can trivially obtain information about the plaintext m_β underlying the challenge ciphertext ct^*_v. Moreover, there is an additional trivial attack **TA3** to obtain information about m_β.

TA3: Via $\mathcal{O}_{\mathrm{ReEnc}}$ queries, $\mathcal{A}$ obtains a chain of re-encryptions $ct^*_v \xrightarrow{\mathcal{O}_{\mathrm{ReEnc}}} ct^{(j_1)}_{v+1} \xrightarrow{\mathcal{O}_{\mathrm{ReEnc}}} \cdots \xrightarrow{\mathcal{O}_{\mathrm{ReEnc}}} ct^{(j_t)}_{v+t}$ starting from the challenge ciphertext ct^*_v and ending at ciphertext $ct_{v+t}^{(j_t)}$ of some corrupted user $j_t \in \mathcal{Q}_c$ from whom $\mathcal{A}$ ever obtains its secret key $sk^{(j_t)}$. In this case, $\mathcal{A}$ can use $sk^{(j_t)}$ to decrypt $ct_{v+t}^{(j_t)}$ to trivially obtain a function of m_β.

To exclude this additional trivial attack, we keep track of a set $\mathcal{L}^*$ to record (index of) the challenge ciphertext ct_v^* as well as all honestly generated re-encryptions of ct_v^* output by $\mathcal{O}_{\text{ReEnc}}$.

3.2 Achieving CPA and HRA Security for Multi-Hop FPRE from Weaker Security Notions: IND, wKP and SH

Our CPA and HRA security for multi-hop FPRE formalized in the previous subsection are defined in an *adaptive* manner, where the adversary $\mathcal{A}$ can designate the challenge user i^* and make all oracle queries adaptively, including corruption queries $\mathcal{O}_{\text{Cor}}$, re-encryption key queries $\mathcal{O}_{\text{ReKey}}$, and honest encryption queries $\mathcal{O}_{\text{Enc}}$ and honest re-encryption queries $\mathcal{O}_{\text{ReEnc}}$ in the case of HRA. Accordingly, the tuples (i, j) for which $\mathcal{A}$ obtains a re-encryption key $\mathsf{rk}_{i\to j}^{f}$ (i.e., the set $\mathcal{Q}_{rk}$ in Fig. 2 and Fig. 3) are adaptively determined by $\mathcal{A}$ and form a complex directed graph. In the case of HRA, the tuples (i, j) for which $\mathcal{A}$ makes a re-encryption query $\mathcal{O}_{\text{ReEnc}}(i, j, \cdot, \cdot)$ form another complex directed graph.

One possible way to achieve adaptive CPA/HRA security is first proving a selective version of CPA/HRA security, and then reducing the adaptive security to the selective counterpart via a guessing strategy. The selective CPA/HRA security means that $\mathcal{A}$ has to declare the graphs for re-encryption keys/re-encryptions at the beginning of the experiment, and thus it is relatively easy to prove selective security in general. However, the price is a considerably large security loss $O(2^{\mathsf{n}^2})$ incurred by the guessing of the graphs.

To reduce the security loss of adaptive security, Jafargholi et al. [15] proposed a generic framework for upgrading selective security to adaptive security with a more fine-grained analysis. Later, Fuchsbauer et al. [12] applied the framework of [15] to the CPA/HRA security of (traditional) PRE.

In this subsection, we will extend the framework of Jafargholi et al. [15] further to the CPA and HRA security of our multi-hop fine-grained PRE, by adapting the techniques of Fuchsbauer et al. [12] to the fine-grained setting. More precisely, we will first defined three weaker security notions, including indistinguishability (IND), weak key-privacy (wKP) and source-hiding (SH), to our multi-hop FPRE, and then establish two theorems showing $\mathsf{CPA}, \mathsf{HRA}$ security of our multi-hop FPRE based on these weaker security notions. The formalization of the weaker security notions and the proofs of the theorems are mainly adapted from [15, 12].

Now we present the formal definitions of IND, wKP, SH for multi-hop FPRE.

Indistinguishability. The IND security of multi-hop FPRE considers the indistinguishability of ciphertexts in a single-user and multi-challenge setting, where the adversary is given no re-encryption keys compared with the CPA security.

Definition 4 (IND **Security).** *A multi-hop FPRE scheme* mFPRE *is* IND *secure, if for any PPT adversary $\mathcal{A}$, it holds that* $\mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$, *where the experiment* $\mathsf{Exp}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{A}}$ *is defined in Fig. 4.*

```
$\mathsf{Exp}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{A}}$:
$(pk, sk) \leftarrow_\$ \mathsf{KGen}$
$\beta \leftarrow_\$ \{0,1\}$
$\beta' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\textsc{Chal}}(\cdot,\cdot,\cdot)}(pk)$
If $\beta' = \beta$: Return 1;   Else: Return 0

$\mathcal{O}_{\textsc{Chal}}(m_0, m_1, v)$:
    $ct_v \leftarrow_\$ \mathsf{Enc}(pk, m_\beta, v)$
    Return $ct_v$
```

Fig. 4. The indistinguishability experiment $\mathsf{Exp}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{A}}$ for mFPRE.

Weak Key-Privacy. The original key-privacy for PREs was introduced in [4]. In [12], weak key-privacy was introduced and it requires the indistinguishability between the re-encryption key $\mathsf{rk}_{0\to j}$ from user 0 to user j and the re-encryption key $\mathsf{rk}_{1\to j}$ from user 1 to user j. Below we adapt it to our multi-hop FPRE, by requiring the existence of a PPT algorithm $\mathsf{FReKGen}^*$ which can simulate the generation of re-encryption keys $\mathsf{rk}^f_{0\to j}$ without the secret key of source user 0.

Definition 5 (wKP **Security).** *A multi-hop FPRE scheme* mFPRE *has weak key privacy (*wKP *security), if there exists a PPT simulation algorithm* $\mathsf{FReKGen}^*$*, s.t. for any PPT adversary $\mathcal{A}$ and any polynomial $\mathfrak{n}$, it holds that* $\mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}} \Rightarrow 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$, *where* $\mathsf{Exp}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ *is defined in Fig. 5.*

```
$\mathsf{Exp}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$:
For $i \in [0,\mathfrak{n}]$:  $(pk^{(i)}, sk^{(i)}) \leftarrow_\$ \mathsf{KGen}$
$\beta \leftarrow_\$ \{0,1\}$
$\beta' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\textsc{ReKey}}(\cdot,\cdot)}(\{pk^{(i)}\}_{i\in[0,\mathfrak{n}]})$
If $\beta' = \beta$: Return 1;   Else: Return 0

$\mathcal{O}_{\textsc{ReKey}}(j \in [\mathfrak{n}], f)$:   //user 0 is always the source user
If $\beta = 0$:                                   //real re-encryption key
    $\mathsf{rk}^f_{0\to j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(0)}, sk^{(0)}, pk^{(j)}, f)$
Else:                                             //simulated re-encryption key
    $\mathsf{rk}^f_{0\to j} \leftarrow_\$ \mathsf{FReKGen}^*(pk^{(0)}, pk^{(j)}, f)$
Returns $\mathsf{rk}^f_{0\to j}$
```

Fig. 5. The weak key-privacy experiment $\mathsf{Exp}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE.

Source-Hiding. Roughly speaking, source-hiding (SH) requires the indistinguishability between freshly-encrypted ciphertexts (via Enc) and re-encrypted ciphertexts (via FReEnc), even if the adversary has all secret keys and re-encryption keys. SH security can help us upgrade CPA security to HRA security for FPRE.

Definition 6 (SH **Security).** *A multi-hop FPRE scheme* mFPRE *has the property of source-hiding (*SH *security), if for any (unbounded) adversary $\mathcal{A}$, it holds that* $\mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$, *where experiment* $\mathsf{Exp}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}}$ *is defined in Fig. 6.*

Achieving CPA and HRA Security for Multi-Hop FPRE. Now we are ready to present two theorems showing (adaptive) CPA and HRA of multi-hop FPRE assuming the weak security notions IND, wKP and SH. The theorems are essentially applications of the framework of Jafargholi et al. [15] and adaptions

```
$\mathsf{Exp}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}}$:
$(pk^{(0)}, sk^{(0)}) \leftarrow\$ \mathsf{KGen}$
$(pk^{(1)}, sk^{(1)}) \leftarrow\$ \mathsf{KGen}$
$\mathcal{Q}_f := \perp$                                  //record functions
$\mathcal{L} := \perp$                //record honestly generated ciphertexts
$\mathsf{ctr} := 0$                   //index of honestly generated ciphertexts
$\beta \leftarrow\$ \{0,1\}$
$\beta' \leftarrow\$ \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot),\mathcal{O}_{\text{Enc}}(\cdot,\cdot),\mathcal{O}_{\text{Chal}}(\cdot,\cdot)}(pk^{(0)}, sk^{(0)}, pk^{(1)}, sk^{(1)})$

If $\beta' = \beta$: Return 1;   Else: Return 0
```

```
$\mathcal{O}_{\text{ReKey}}(f)$:   //re-key from user 0 to user 1 on function $f$
$\mathsf{rk}^f_{0\to1} \leftarrow\$ \mathsf{FReKGen}(pk^{(0)}, sk^{(0)}, pk^{(1)}, f)$
$\mathcal{Q}_f := \mathcal{Q}_f \cup \{f\}$
Return $\mathsf{rk}^f_{0\to1}$
```

```
$\mathcal{O}_{\text{Enc}}(m, v)$:   //honestly generated ciphertext of user 0
$\mathsf{ctr} := \mathsf{ctr} + 1$
$ct^{(0)}_v \leftarrow\$ \mathsf{Enc}(pk^{(0)}, m, v)$
$\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, m, (ct^{(0)}_v, v))\}$
Return $(\mathsf{ctr}, ct^{(0)}_v)$
```

```
$\mathcal{O}_{\text{Chal}}(k, f)$:                          //challenge oracle
Retrieve $(k, m, (ct^{(0)}_v, v))$ from $\mathcal{L}$:
    If fails, return $\perp$
If $\beta = 0$:                                 //re-encrypted ciphertext
    If $f \notin \mathcal{Q}_f$: $\mathsf{rk}^f_{0\to1} \leftarrow\$ \mathsf{FReKGen}(pk^{(0)}, sk^{(0)}, pk^{(1)}, f)$
    $ct^{(1)}_{v+1} \leftarrow\$ \mathsf{FReEnc}(\mathsf{rk}^f_{0\to1}, ct^{(0)}_v, v)$
Else:                                       //freshly-encrypted ciphertext
    $ct^{(1)}_{v+1} \leftarrow\$ \mathsf{Enc}(pk^{(1)}, f(m), v+1)$
Return $ct^{(1)}_{v+1}$
```

Fig. 6. The source-hiding experiment $\mathsf{Exp}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}}$ for mFPRE.

of the techniques of Fuchsbauer et al. [12] to multi-hop FPRE. We postpone the proofs of the theorems to Appendix B.2 and Appendix B.3 respectively, as they almost verbatim follow [12, 15].

To state the theorems precisely, we consider an adversary $\mathcal{A}$ in the CPA/HRA security experiment, and define some notations. If we view users $[\mathfrak{n}]$ as vertices and re-encryption keys $\mathsf{rk}^f_{i\to j}$ that $\mathcal{A}$ obtains through $\mathcal{O}_{\text{ReKey}}$ queries as an edge from i to j, then it forms a directed graph. We define the subgraph that is reachable from the challenge user i^* as *the challenge graph* of $\mathcal{A}$, denoted by G. For the challenge graph G, if we denote by δ the outdegree (i.e., the maximum outdegree over all vertices) and d the depth, then the challenge graph is in the graph class $\mathcal{G}(\mathfrak{n}, \delta, d)$ of all graphs with $\mathfrak{n}$ vertices, outdegree δ and depth d.

In Appendix B.1, we further define the pebbling time complexity τ and space complexity σ for the class $\mathcal{G}(\mathfrak{n}, \delta, d)$, respectively, according to [12, 15].

Theorem 1 (IND + wKP ⇒ CPA for Multi-Hop FPRE). *If a multi-hop FPRE scheme* mFPRE *has both* IND *and* wKP *security, then it is* CPA *secure.*

More precisely, for any PPT adversary $\mathcal{A}$ against the CPA *security with challenge graph G in $\mathcal{G}(\mathfrak{n}, \delta, d)$ whose pebbling time complexity is τ and space complexity is σ, there exist PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ s.t.* $\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) \leq (2 \cdot \mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{B}} + 2\tau \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}) \cdot \mathfrak{n}^{\sigma+\delta+1}$. *We refer to Appendix B.1 for the definitions of pebbling time complexity τ and space complexity σ.*

Theorem 2 (IND + wKP + SH ⇒ HRA for Multi-Hop FPRE). *If a multi-hop FPRE scheme* mFPRE *has* IND, wKP *and* SH *security simultaneously, then it is* HRA *secure.*

More precisely, for any PPT adversary $\mathcal{A}$ against the HRA *security with challenge graph G in $\mathcal{G}(\mathfrak{n}, \delta, d)$ whose pebbling time complexity is τ and space complexity is σ, there exist PPT algorithms $\mathcal{B}, \mathcal{B}'$ and $\mathcal{B}''$ s.t.* $\mathsf{Adv}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) \leq (2 \cdot \mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{B}} + 2\tau \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}) \cdot \mathfrak{n}^{\sigma+\delta+1} + 2\mathfrak{n}(\mathfrak{n}-1)L \cdot \mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}''}$*, where L is the maximum level supported by* mFPRE. [5] *We refer to Appendix B.1 for the definitions of pebbling time complexity τ and space complexity σ.*

[5] We note that our Theorem 2 has slightly different parameters than the corresponding theorem (i.e., Theorem 6) in [12]. Jumping ahead, this is because we use slightly

Note that the security loss of Theorem 1 and Theorem 2 is dominating by $2\tau \cdot \mathfrak{n}^{\sigma+\delta+1}$ and $2\mathfrak{n}(\mathfrak{n}-1)L$.

- For an arbitrary adversary $\mathcal{A}$ with an arbitrary challenge graph G, according to the bounds given in [12] (cf. Lemma 6 in Appendix B.1), we have the pebbling time complexity $\tau \leq (2\delta)^d$, the space complexity $\sigma \leq \mathfrak{n}$, the outdegree $\delta \leq \mathfrak{n}$ and the depth $d \leq \mathfrak{n}$. Moreover, L is (at most) a polynomial in $\mathfrak{n}$. Consequently, the security loss for arbitrary adversary $\mathcal{A}$ is $\mathfrak{n}^{O(\mathfrak{n})}$.
- In many realistic scenarios like key rotation for encrypted cloud storage or forwarding of encrypted mail, as demonstrated in [12], the proxy relations are in fact *trees, chains or low-depth graphs*, so does the challenge graph G. In these situations, according to the bounds given in [12] (cf. Lemma 6 in Appendix B.1), we have the pebbling time complexity $\tau = O(1)^{\log \mathfrak{n}}$, the space complexity $\sigma = O(\log \mathfrak{n})$ and the outdegree $\delta =$ constant, and consequently, the security loss is only quasi-polynomial $\mathfrak{n}^{O(\log \mathfrak{n})}$.

3.3 Other Security Notions for Multi-Hop FPRE: UNID and CUL

In this subsection, we formalize two additional security notions for multi-hop FPRE, namely unidirectionality (UNID) and ciphertext unlinkability (CUL), by adapting the formalization in [24] defined for single-hop FPRE.

Unidirectionality. Intuitively, unidirectionality (UNID) means that the proxy ability in one direction does not imply the proxy ability in the other direction. More precisely, it requires that given a re-encryption key $\mathsf{rk}^{f}_{j^*\to i^*}$, it is hard for an adversary to come up with re-encryption key $\mathsf{rk}^{f'}_{i^*\to j^*}$ of the other direction even if the adversary is able to obtain some re-encryption keys and corrupt some users to obtain their secret keys. The formal definition is as follows.

Definition 7 (Unidirectionality for Multi-Hop FPRE). *A multi-hop FPRE scheme* mFPRE *is unidirectional (*UNID *secure), if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $\mathfrak{n}$*, it holds that* $\mathsf{Adv}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ *is defined in Fig. 7.*

In Appendix E.1, we give some explanations of the UNID security definition and discuss the trivial attacks **TA1′-TA5′** in Remark 4, and then show that the UNID security is implied by the CPA security in Lemma 10 for multi-hop FPRE.

Ciphertext Unlinkability. In real scenarios, re-encryption relations between ciphertexts often imply the proxy connections between users. Therefore, it is

different proof strategy than [12] when reducing to SH, in order to change all re-encrypted ciphertexts to freshly generated ciphertexts: in [12], they change a pair of re-encrypted ciphertexts at a time, resulting in the factor $(Q_{\mathsf{E}} + Q_{\mathsf{RE}}) \cdot Q_{\mathsf{RE}}$ (i.e., the number of ciphertext pairs); in contrast, we change all re-encrypted ciphertexts in one layer at a time, and layer by layer, resulting in the factor L (i.e., the maximum number of layers). We refer to Fig. 12 and Fig. 13 in Appendix B.3 for an illustration of our strategy.

```
$\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$:
For $i \in [\mathfrak{n}]$:  $(pk^{(i)}, sk^{(i)}) \leftarrow_\$ \mathsf{KGen}$
$\mathcal{Q}_{rk} := \emptyset$                      //record re-encryption key queries
$\mathcal{Q}_c := \emptyset$                            //record corruption queries
$i^* := \perp, j^* := \perp$                          //record challenge users
$(i^*, j^*, f, st) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{Cor}}(\cdot)}(\{pk^{(i)}\}_{i\in[\mathfrak{n}]})$
If $(i^* = j^*)$ or $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$:
    Return 0                   //avoid $\mathbf{TA1}', \mathbf{TA2}', \mathbf{TA3}', \mathbf{TA4}'$
$\mathsf{rk}^f_{j^* \to i^*} \leftarrow_\$ \mathsf{FReKGen}(pk^{(j^*)}, sk^{(j^*)}, pk^{(i^*)}, f)$
$\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(j^*, i^*)\}$
$(f', \mathsf{rk}^{f'}_{i^* \to j^*}) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{Cor}}(\cdot)}(st, \mathsf{rk}^f_{j^* \to i^*})$
If $f'$ does not have output diversity:
    Return $\perp$                                    //avoid $\mathbf{TA5}'$
//check the functionality of $\mathsf{rk}^{f'}_{i^* \to j^*}$ in the following way
$m \leftarrow_\$ \mathcal{M}$, $ct_0^{(i^*)} \leftarrow_\$ \mathsf{Enc}(pk^{(i^*)}, m, 0)$
$ct_1^{(j^*)} \leftarrow_\$ \mathsf{FReEnc}(\mathsf{rk}^{f'}_{i^* \to j^*}, ct_0^{(i^*)}, 0)$
If $\mathsf{Dec}(sk^{(j^*)}, ct_1^{(j^*)}) = f'(m)$:
    Return 1
Else:  Return 0
```

```
$\mathcal{O}_{\text{ReKey}}(i, j, f)$:
    If $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$:
        Return $\perp$                //avoid $\mathbf{TA3}', \mathbf{TA4}'$
    $\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(i,j)\}$
    $\mathsf{rk}^f_{i \to j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    Return $\mathsf{rk}^f_{i \to j}$

$\mathcal{O}_{\text{Cor}}(i)$:
    If $i = i^*$:  Return $\perp$                //avoid $\mathbf{TA2}'$
    If $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$:
        Return $\perp$                //avoid $\mathbf{TA3}', \mathbf{TA4}'$
    $\mathcal{Q}_c := \mathcal{Q}_c \cup \{i\}$
    Return $sk^{(i)}$

$\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk}, \mathcal{Q}_c)$:   //avoid $\mathbf{TA3}', \mathbf{TA4}'$
    If $\exists\ (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$
        s.t. $(j_t \in \mathcal{Q}_c)$ or $(j_t = j^*)$ for some $t \geq 1$:
        Return 1
    Else:  Return 0
```

Fig. 7. The Unidirectionality security experiment $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE, where "output diversity" is defined as $\Pr[m_0, m_1 \leftarrow_\$ \mathcal{M} : f'(m_0) \neq f'(m_1)] \geq 1/\mathsf{poly}(\lambda)$ (see Remark 4 in Appendix E.1 for more details).

desirable to hide the relations/connections, which is captured by the property ciphertext unlinkability (CUL). We formalize CUL for multi-hop FPRE by requiring the indistinguishability between a chain of re-encrypted ciphertexts

$$ct_0^{(i_0)} \xrightarrow{\mathsf{rk}^{f_1}_{i_0 \to i_1}} ct_1^{(i_1)} \xrightarrow{\mathsf{rk}^{f_2}_{i_1 \to i_2}} \cdots \xrightarrow{\mathsf{rk}^{f_L}_{i_{L-1} \to i_L}} ct_L^{(i_L)}$$

generated by FReEnc and a set of freshly and independently encrypted ciphertexts $(ct_0^{(i_0)}, ct_1^{(i_1)}, \ldots, ct_L^{(i_L)})$ generated by Enc.

Definition 8 (Ciphertext Unlinkability for Multi-Hop PRE). *A multi-hop FPRE scheme* mFPRE *has ciphertext unlinkability* (CUL)*, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $\mathfrak{n}$*, it holds that* $\mathsf{Adv}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) := \big|\Pr[\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}} \Rightarrow 1] - \frac{1}{2}\big| \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ *is defined in Fig. 8.*

In Appendix E.2, we give some explanations of the CUL security definition and discuss the trivial attacks **TA1″-TA2″** in Remark 5. We note that CUL security is similar to the SH security (cf. Def. 6) as they both capture the indistinguishability of re-encrypted ciphertexts and freshly generated ciphertexts. However, CUL security is defined in a much more realistic setting compared with the SH security: CUL considers a setting of multiple users while SH deals with only two users, and moreover, CUL protects the unlinkability of a chain of L re-encrypted ciphertexts with L the maximum level of mFPRE, while SH considers only chains of two ciphertexts. Nevertheless, in Appendix E.2, we will show that the CUL security is implied by the SH + CPA security in Lemma 11.

Remark 3 (Post-Compromise Security). In [10], Davidson et al. proposed post-compromise security (PCS) for PRE, which considers the scenario where PRE

```
$\underline{\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}}$:
For $i \in [\mathfrak{n}]$: $(pk^{(i)}, sk^{(i)}) \leftarrow_\$ \mathsf{KGen}$
$\mathcal{Q}_{rk} := \emptyset$                                   //record re-encryption key queries
$\mathcal{Q}_c := \emptyset$                                      //record corruption queries
$\mathcal{Q}_u := \emptyset$                                      //record challenge users
$\left(\{i_j\}_{j\in[0,L]}, \begin{pmatrix}(\{f_j\}_{j\in[L]}, m)\\(m_0, m_1, \ldots, m_L)\end{pmatrix}, st\right) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\textsc{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\textsc{Cor}}(\cdot)}\left(\{pk^{(i)}\}_{i\in[\mathfrak{n}]}\right)$
$\mathcal{Q}_u := \{i_j\}_{j\in[0,L]}$                             //update challenge users
If $(\exists j \in [0, L]$ s.t. $i_j \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$:
    Return $b \leftarrow_\$ \{0,1\}$                                //avoid $\mathbf{TA1}''$, $\mathbf{TA2}''$
$\beta \leftarrow_\$ \{0,1\}$
If $\beta = 0$:
    $ct_0^{(i_0)} \leftarrow_\$ \mathsf{Enc}(pk^{(i_0)}, m, 0)$
    For $j \in [L]$:                                         //re-encrypted ciphertexts
        $\mathsf{rk}^{f_j}_{i_{j-1}\to i_j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i_{j-1})}, sk^{(i_{j-1})}, pk^{(i_j)}, f_j)$
        $ct_j^{(i_j)} \leftarrow_\$ \mathsf{FReEnc}(\mathsf{rk}^{f_j}_{i_{j-1}\to i_j}, ct^{(i_{j-1})}, j-1)$
If $\beta = 1$ :
    For $j \in [0, L]$:                                      //independently generated ciphertexts
        $ct_j^{(i_j)} \leftarrow_\$ \mathsf{Enc}(pk^{(i_j)}, m_j, j)$
$\beta' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\textsc{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\textsc{Cor}}(\cdot)}(st, \{\mathsf{rk}^{f_j}_{i_{j-1}\to i_j}\}_{j\in[L]}, \{ct_j^{(i_j)}\}_{j\in[0,L]})$
If $\beta' = \beta$: Return 1;   Else: Return 0
```

```
$\underline{\mathcal{O}_{\textsc{ReKey}}(i, j, f)}$:   //re-encryption key queries
    If $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$:
        Return $\perp$                          //avoid $\mathbf{TA2}''$
    $\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(i,j)\}$
    $\mathsf{rk}^f_{i\to j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    Return $\mathsf{rk}^f_{i\to j}$

$\underline{\mathcal{O}_{\textsc{Cor}}(i)}$:                         //corruption queries
    If $i \in \mathcal{Q}_u$:  Return $\perp$          //avoid $\mathbf{TA1}''$
    If $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$:
        Return $\perp$                          //avoid $\mathbf{TA2}''$
    $\mathcal{Q}_c := \mathcal{Q}_c \cup \{i\}$
    Return $sk^{(i)}$

$\underline{\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c)}$:          //check $\mathbf{TA2}''$
    If $\exists\ i^* \in \mathcal{Q}_u$ and
        $\exists\ (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$
        s.t. $j_t \in \mathcal{Q}_c$ for some $t \geq 1$:
        Return 1
    Else: Return 0
```

Fig. 8. The Ciphertext Unlinkability security experiment $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE.

serves for key rotation and guarantees that security still exists after the compromise of past secret keys. More concretely, suppose that Alice has stored some encrypted data and wants to update her public key from pk to pk'. To this end, she can generate an update token (i.e., a re-encryption key from pk to pk'), and re-encrypts the encrypted data using the token. In such scenario, PCS ensures that an adversary cannot distinguish which of two adversarially-chosen ciphertexts a re-encryption was created from, even when given the old secret key (i.e., the sk corresponding to pk) and the update token. Davidson et al. [10] also discussed the relations between PCS and other security notions of PRE, and proved that HRA together with SH imply PCS for (non-fine-grained) PRE.

Following their work [10], we can extend PCS for our multi-hop FPRE, by requiring the indistinguishability between fine-grained re-encryptions of two adversarially chosen ciphertexts, even if the adversary can obtain the old secret key and the fine-grained re-encryption key used to perform the re-encryption. Moreover, similar to [10], we can also show that $\mathsf{HRA} + \mathsf{SH} \Rightarrow$ PCS holds for our multi-hop FPRE. The formalization of PCS and the proof of $\mathsf{HRA} + \mathsf{SH} \Rightarrow$ PCS for multi-hop FPRE are straightforward based on [10], and we will not elaborate on them. Jumping ahead, our multi-hop FPRE scheme mFPRE_2 in Subsect. 4.2 is both HRA and SH secure, and thus achieves PCS.

4 Constructions of Multi-Hop Fine-Grained PRE Scheme

In this section, we present two constructions of multi-hop fine-grained PRE (mFPRE) schemes, including a CPA secure scheme mFPRE_1 and an HRA secure scheme mFPRE_2, from the LWE assumptions.

4.1 The CPA secure Multi-Hop FPRE Scheme mFPRE$_1$

Parameters. Let $\mathsf{pp}_{\mathsf{LWE}} = (p, q, n, N, L, \ell, \gamma, \Delta, \chi)$ be LWE-related parameters that meet the following conditions:

- $p, q, n, N, L, \ell, \gamma, \Delta \in \mathbb{N}$ are integers, where $q := p^2$, $\gamma \geq O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$;
- χ is a B-bounded distribution, where B satisfies $\gamma \cdot \omega(\log n) \leq B < \min\{p/2, q/(10N)\}$ and $(nB + NB + \ell\Delta)^L B < \min\{p/2, q/(10N)\}$.

More precisely, we describe two settings of parameter in Table 2, one for constant hops ($L = c$) and under polynomial modulus q, while the other for sub-linear hops ($L = c \cdot \sqrt[3]{\lambda}$) under sub-exponential modulus q. For simplicity, we assume that all algorithms of our scheme mFPRE$_1$ take $\mathsf{pp}_{\mathsf{LWE}}$ as an implicit input.

Table 2. Concrete parameters setting, where λ denotes the security parameter and c denotes an arbitrary constant.

Parameters	p	q	n	N	L	ℓ	γ	Δ	B
Settings (L = constant)	λ^{2c+1}	λ^{4c+2}	λ	λ	c	λ	$\sqrt{\lambda}(\log\lambda)^2$	λ	$\sqrt{\lambda}(\log\lambda)^4$
Settings (L = sub-linear)	$2^{\sqrt{\lambda}}$	$2^{2\sqrt{\lambda}}$	λ	λ	$c \cdot \sqrt[3]{\lambda}$	λ	$\sqrt{\lambda}(\log\lambda)^2$	λ	$\sqrt{\lambda}(\log\lambda)^4$

Bounded Linear function family. The message space is $\mathcal{M} := \mathbb{Z}_p^\ell$. Define the family of bounded linear functions $\mathcal{F}_{\text{lin}}$ from $\mathcal{M}$ to $\mathcal{M}$ over $\mathbb{Z}_p$ as follows:

$$\mathcal{F}_{\text{lin}} = \left\{ \begin{array}{c} f_{\mathbf{M}} : \mathbb{Z}_p^\ell \to \mathbb{Z}_p^\ell \\ \mathbf{m} \mapsto \mathbf{M} \cdot \mathbf{m} \bmod p \end{array} \,\middle|\, \mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell}, \|\mathbf{M}\|_\infty \leq \Delta \right\}. \tag{9}$$

LWE-based Multi-Hop FPRE Scheme mFPRE$_1$. Let TrapGen, SamplePre, Invert be the PPT algorithms defined in Lemmas 1, 2 and 3 in Appendix A.2, respectively. Our LWE-based multi-hop FPRE scheme mFPRE$_1$ = (KGen, FReKGen, Enc, FReEnc, Dec) for the bounded linear function family $\mathcal{F}_{\text{lin}}$ (9) is shown in Fig. 9.

```
$(pk, sk) \leftarrow_\$ \mathsf{KGen}$:
$(\overline{\mathbf{A}} \in \mathbb{Z}_q^{N \times n}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, 1^N)$
$\underline{\mathbf{A}} \leftarrow_\$ \mathbb{Z}_q^{\ell \times n}$
$pk := \mathbf{A} = \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}} \in \mathbb{Z}_q^{(N+\ell) \times n}$
$sk := \mathbf{T}$
Return $(pk, sk)$

$\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i)} = \mathbf{A}^{(i)}, sk^{(i)} = \mathbf{T}^{(i)}, pk^{(j)} = \mathbf{A}^{(j)}, f_{\mathbf{M}} \in \mathcal{F}_{\text{lin}})$:
$\mathbf{S} \leftarrow_\$ \chi^{n \times n}$, $\mathbf{E} \leftarrow_\$ \chi^{(N+\ell) \times n}$
Parse $\mathbf{A}^{(i)} = \binom{\overline{\mathbf{A}}^{(i)}}{\underline{\mathbf{A}}^{(i)}}$
$\mathbf{R} \in \mathbb{Z}^{(N+\ell) \times N} \leftarrow_\$ \mathsf{SamplePre}\left(\mathbf{T}^{(i)}, \overline{\mathbf{A}}^{(i)}, \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(i)}, \gamma\right)$
$\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} := \left(\mathbf{R} \,\middle|\, \begin{matrix}\mathbf{0} \\ \mathbf{M}\end{matrix}\right) \in \mathbb{Z}_p^{(N+\ell) \times (N+\ell)}$    //$\mathbf{M}$ is the description of $f_{\mathbf{M}}$
Return $\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}}$

$ct_v \leftarrow_\$ \mathsf{Enc}(pk = \mathbf{A}, \mathbf{m} \in \mathcal{M}, v \in [0, L])$:
$\mathbf{s} \leftarrow_\$ \chi^n$, $\mathbf{e} \leftarrow_\$ \chi^{N+\ell}$
$ct_v := \mathbf{A}\mathbf{s} + \mathbf{e} + \binom{\mathbf{0}}{p\mathbf{m}} \in \mathbb{Z}_q^{N+\ell}$

$ct_{v+1}^{(j)} \leftarrow \mathsf{FReEnc}(\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \in \mathbb{Z}_p^{(N+\ell) \times (N+\ell)},$
        $ct_v^{(i)} \in \mathbb{Z}_q^{N+\ell}, v \in [0, L-1])$:
$ct_{v+1}^{(j)} := \mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \cdot ct_v^{(i)} \in \mathbb{Z}_q^{N+\ell}$
Return $ct_{v+1}^{(j)}$

$\mathbf{m} \leftarrow \mathsf{Dec}(sk = \mathbf{T}, ct \in \mathbb{Z}_q^{N+\ell})$:
Parse $ct = \binom{\overline{ct} \in \mathbb{Z}_q^N}{\underline{ct} \in \mathbb{Z}_q^\ell}$
$(\mathbf{s}, \overline{\mathbf{e}}) \leftarrow_\$ \mathsf{Invert}(\mathbf{T}, \overline{ct})$
$\tilde{\mathbf{m}} = (\tilde{m}_1, \ldots, \tilde{m}_\ell) := \underline{ct} - \underline{\mathbf{A}}\mathbf{s}$
For $i \in [\ell]$:  $m_i := \lceil \tilde{m}_i / p \rfloor$
Return $\mathbf{m} = (m_1, m_2, \ldots, m_\ell)$
```

Fig. 9. The LWE-based Multi-Hop FPRE scheme mFPRE$_1$ for $\mathcal{F}_{\text{lin}}$.

Correctness. Let $pk = \mathbf{A}$ and $sk = \mathbf{T}$. For a v-level ciphertext ct_v generated by $\mathsf{Enc}(pk, \mathbf{m}, v)$, we have $ct_v = \binom{\overline{ct_v}}{\underline{ct_v}} = \binom{\overline{\mathbf{A}}\mathbf{s}+\overline{\mathbf{e}}}{\underline{\mathbf{A}}\mathbf{s}+\underline{\mathbf{e}}+p\mathbf{m}}$, where $\mathbf{e} = \binom{\overline{\mathbf{e}}}{\underline{\mathbf{e}}} \leftarrow_{\$} \chi^{N+\ell}$ and the upper part is an LWE instance of $\overline{\mathbf{A}}$. Since $\overline{\mathbf{e}}$ is B-bounded with $B < q/(10N)$, $\|\overline{\mathbf{e}}\| \leq \sqrt{N}\, \|\overline{\mathbf{e}}\|_\infty \leq \sqrt{N}B < q/(10\sqrt{N})$. Then by Lemma 2 in Appendix A.2, $(\mathbf{s}, \overline{\mathbf{e}})$ can be correctly recovered via $(\mathbf{s}, \overline{\mathbf{e}}) \leftarrow \mathsf{Invert}(\mathbf{T}, \overline{ct_v})$. Thus according to the decryption algorithm $\mathsf{Dec}(sk, ct_v)$, we get $\tilde{\mathbf{m}} = \underline{ct_v} - \underline{\mathbf{A}}\mathbf{s} = \underline{\mathbf{e}} + p\mathbf{m}$, and by parsing $\underline{\mathbf{e}} = (e_1, \ldots, e_\ell)^\top$, we have that $\tilde{m}_i = e_i + pm_i$ for all $i \in [\ell]$. Moreover, since $\underline{\mathbf{e}}$ is B-bounded with $B < p/2$, each $|e_i| \leq B < p/2$. Consequently, $\lceil \tilde{m}_i/p \rfloor = m_i$ and Dec can recover $\mathbf{m}$ correctly from ct_v.

Fine-Grained L-Hop Correctness. For $ct_0^{(i)} \xrightarrow{\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}}} ct_1^{(j)}$, where $ct_0^{(i)} \leftarrow_{\$} \mathsf{Enc}(pk^{(i)}, \mathbf{m}, 0)$, $\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f_{\mathbf{M}_1})$ and $ct_1^{(j)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}}, ct_0^{(i)}, 0)$, we will show that the decryption of $ct_1^{(j)}$ results in $f_{\mathbf{M}_1}(\mathbf{m}) = \mathbf{M}_1\mathbf{m}$. More precisely, let $\mathsf{rk}_{i\to j}^{\mathbf{M}_1} := \left(\mathbf{R}_1 \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}_1\end{smallmatrix}\right)$, we have

$$
\begin{aligned}
ct_1^{(j)} &:= \left(\mathbf{R}_1 \,\middle|\, \begin{matrix}\mathbf{0}\\ \mathbf{M}_1\end{matrix}\right) \cdot ct_0^{(i)} = \left(\mathbf{R}_1 \,\middle|\, \begin{matrix}\mathbf{0}\\ \mathbf{M}_1\end{matrix}\right) \cdot \left(\begin{pmatrix}\overline{\mathbf{A}}^{(i)}\\ \underline{\mathbf{A}}^{(i)}\end{pmatrix}\mathbf{s}_0 + \begin{pmatrix}\overline{\mathbf{e}_0}\\ \underline{\mathbf{e}_0}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ p\mathbf{m}\end{pmatrix}\right) \\
&= \left(\mathbf{R}_1\overline{\mathbf{A}}^{(i)} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\end{pmatrix}\underline{\mathbf{A}}^{(i)}\right) \cdot \mathbf{s}_0 + \mathbf{R}_1\overline{\mathbf{e}_0} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\underline{\mathbf{e}_0}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ p \cdot \mathbf{M}_1\mathbf{m}\end{pmatrix} \\
&= \left(\mathbf{A}^{(j)}\mathbf{S} + \mathbf{E}\right) \cdot \mathbf{s}_0 + \mathbf{R}_1\overline{\mathbf{e}_0} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\underline{\mathbf{e}_0}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ p \cdot \mathbf{M}_1\mathbf{m}\end{pmatrix} \\
&= \mathbf{A}^{(j)} \underbrace{\mathbf{S}\mathbf{s}_0}_{:=\mathbf{s}_1} + \underbrace{\mathbf{E}\mathbf{s}_0 + \mathbf{R}_1\overline{\mathbf{e}_0} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\underline{\mathbf{e}_0}\end{pmatrix}}_{:=\mathbf{e}_1} + \begin{pmatrix}\mathbf{0}\\ p \cdot \underbrace{\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_1}(\mathbf{m})}\end{pmatrix},
\end{aligned}
\tag{10}
$$

where $\mathbf{s}_0 \leftarrow_{\$} \chi^n$, $\mathbf{e}_0 = \binom{\overline{\mathbf{e}_0}}{\underline{\mathbf{e}_0}} \leftarrow_{\$} \chi^{N+\ell}$, $\mathbf{S} \leftarrow_{\$} \chi^{n\times n}$, $\mathbf{E} \leftarrow_{\$} \chi^{(N+\ell)\times n}$. Here the second last equality follows from the fact that $\mathbf{R}_1$ generated by $\mathbf{R}_1 \leftarrow_{\$} \mathsf{SamplePre}(\mathbf{T}^{(i)}, \overline{\mathbf{A}}^{(i)}, \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \binom{\mathbf{0}}{\mathbf{M}_1}\underline{\mathbf{A}}^{(i)}, \gamma)$ satisfies $\mathbf{R}_1\overline{\mathbf{A}}^{(i)} = \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \binom{\mathbf{0}}{\mathbf{M}_1}\underline{\mathbf{A}}^{(i)}$ and $\|\mathbf{R}_1\|_\infty \leq \gamma \cdot \omega(\log n)$ according to Lemma 3 in Appendix A.2. Besides, $\|\mathbf{R}_1\|_\infty \leq \gamma \cdot \omega(\log n)$ implies that $\|\mathbf{R}_1\|_\infty \leq B$ due to $\gamma \cdot \omega(\log n) \leq B$. Now that $\mathbf{S}, \mathbf{E}, \mathbf{R}_1, \mathbf{s}_0, \mathbf{e}_0$ are all B-bounded and $\mathbf{M}_1$ is $\varDelta$-bounded, so we have $\|\mathbf{s}_1\|_\infty \leq nB^2$ and $\|\mathbf{e}_1\|_\infty \leq (nB + NB + \ell\varDelta)B < \min\{p/2, q/(10N)\}$. Then by a similar argument as that for correctness, since $\|\mathbf{e}_1\|_\infty < q/(10N)$ and $\|\mathbf{e}_1\|_\infty < p/2$, the decryption algorithm Dec recovers $f_{\mathbf{M}_1}(\mathbf{m}) = \mathbf{M}_1\mathbf{m}$ from $ct_1^{(j)}$.

Next suppose that $ct_1^{(j)}$ is further re-encrypted to $ct_2^{(k)}$, i.e., $ct_1^{(j)} \xrightarrow{\mathsf{rk}_{j\to k}^{f_{\mathbf{M}_2}}} ct_2^{(k)}$, where $\mathsf{rk}_{j\to k}^{f_{\mathbf{M}_2}} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(j)}, sk^{(j)}, pk^{(k)}, f_{\mathbf{M}_2})$ and $ct_2^{(k)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}_{j\to k}^{f_{\mathbf{M}_2}}, ct_1^{(j)}, 1)$, we will show that the decryption of $ct_2^{(k)}$ results in $f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m})) = \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{m}$.

By a similar analysis as above, let $\mathsf{rk}_{j\to k}^{f_{\mathbf{M}_2}} := \left(\mathbf{R}_2 \mid {}^{\mathbf{0}}_{\mathbf{M}_2}\right)$, we have

$$ct_2^{(k)} := \left(\mathbf{R}_2 \;\middle|\; \begin{matrix}\mathbf{0}\\ \mathbf{M}_2\end{matrix}\right) \cdot ct_1^{(j)} = \left(\mathbf{R}_2 \;\middle|\; \begin{matrix}\mathbf{0}\\ \mathbf{M}_2\end{matrix}\right) \cdot \left(\begin{pmatrix}\overline{\mathbf{A}}^{(j)}\\ \underline{\mathbf{A}}^{(j)}\end{pmatrix}\mathbf{s}_1 + \begin{pmatrix}\overline{\mathbf{e}_1}\\ \underline{\mathbf{e}_1}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ p\mathbf{M}_1\mathbf{m}\end{pmatrix}\right)$$

$$= \mathbf{A}^{(k)} \underbrace{\mathbf{S}\mathbf{s}_1}_{:=\mathbf{s}_2} + \underbrace{\mathbf{E}\mathbf{s}_1 + \mathbf{R}_2\overline{\mathbf{e}_1} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_2\underline{\mathbf{e}_1}\end{pmatrix}}_{:=\mathbf{e}_2} + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_2\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m}))}\end{pmatrix},$$

where $\mathbf{S} \leftarrow_\$ \chi^{n\times n}$ and $\mathbf{E} \leftarrow_\$ \chi^{(N+\ell)\times n}$. Similarly, we know that $\mathbf{S}, \mathbf{E}, \mathbf{R}_2$ are B-bounded and $\mathbf{M}_2$ is Δ-bounded. Together with the fact that $\|\mathbf{s}_1\|_\infty \le nB^2 \le (nB + NB + \ell\Delta)B$ and $\|\mathbf{e}_1\|_\infty \le (nB + NB + \ell\Delta)B$, it follows that $\|\mathbf{s}_2\|_\infty \le (nB + NB + \ell\Delta)nB^2$ and $\|\mathbf{e}_2\|_\infty \le (nB + NB + \ell\Delta)^2B < \min\{p/2, q/(10N)\}$. Again, with a similar argument as that for correctness, the decryption algorithm Dec recovers $f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m})) = \mathbf{M}_2\mathbf{M}_1\mathbf{m}$ from $ct_2^{(k)}$.

As the re-encryption proceeds, after L hops of re-encryption under $f_{\mathbf{M}_1}, f_{\mathbf{M}_2}, \cdots, f_{\mathbf{M}_L}$, we get an L-level ciphertext $ct_L^{(\eta)}$ and it satisfies

$$ct_L^{(\eta)} = \mathbf{A}^{(\eta)}\mathbf{s}_L + \mathbf{e}_L + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_L\cdots\mathbf{M}_2\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_L}(\cdots f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m})))}\end{pmatrix},$$

where $\|\mathbf{s}_L\|_\infty \le (nB + NB + \ell\Delta)^{L-1}nB^2$ and $\|\mathbf{e}_L\|_\infty \le (nB + NB + \ell\Delta)^LB < \min\{p/2, q/(10N)\}$. Consequently, the function value $f_{\mathbf{M}_L}(\cdots f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m}))) = \mathbf{M}_L\cdots\mathbf{M}_2\mathbf{M}_1\mathbf{m}$ can be recovered from $ct_L^{(\eta)}$ by the decryption algorithm Dec.

Below we show the IND security and wKP security of our scheme mFPRE_1 via the following two theorems. Then together with Theorem 1 (IND + wKP $\Rightarrow$ CPA) in Subsect. 3.2, it yields the CPA security of our scheme mFPRE_1.

Theorem 3 (IND Security of mFPRE_1). *Assume that the $\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption holds, then the scheme mFPRE_1 proposed in Fig. 9 has IND security. More precisely, for any PPT adversary $\mathcal{A}$ that make at most Q_{chal} queries to $\mathcal{O}_{\text{CHAL}}$, there exists a PPT algorithm $\mathcal{B}$ against the LWE assumption s.t.* $\mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE}_1,\mathcal{A}}(\lambda) \le Q_{chal} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)$.

Proof of Theorem 3. We prove the theorem via two games G_0 and G_1.

Game G_0: This is the IND experiment (cf. Fig. 4). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE}_1,\mathcal{A}}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Let $(pk = \mathbf{A}, sk = \mathbf{T})$. In this game, the challenger chooses a random bit $\beta \leftarrow_\$ \{0,1\}$ and answers $\mathcal{A}$'s $\mathcal{O}_{\text{CHAL}}$ queries $(\mathbf{m}_0, \mathbf{m}_1, v)$ with $ct_v \leftarrow_\$ \mathsf{Enc}(pk, \mathbf{m}_\beta, v)$, i.e., $ct_v := \mathbf{A}\mathbf{s} + \mathbf{e} + \binom{\mathbf{0}}{p\mathbf{m}_\beta}$ for $\mathbf{s} \leftarrow_\$ \chi^n, \mathbf{e} \leftarrow_\$ \chi^{N+\ell}$.

Game G_1: It is the same as G_0, except that, when answering $\mathcal{O}_{\text{CHAL}}(\mathbf{m}_0, \mathbf{m}_1, v)$ queries, the challenger returns a uniformly sampled $ct_v \leftarrow_\$ \mathbb{Z}_q^{N+\ell}$ to $\mathcal{A}$. Clearly, now the challenge bit β is completely hidden to $\mathcal{A}$, thus $\Pr_1[\mathsf{Win}] = \frac{1}{2}$.

It is not hard to see that the $ct_v \leftarrow_\$ \mathsf{Enc}(pk, \mathbf{m}_\beta, v)$ in G_0 is computationally indistinguishable from the $ct_v \leftarrow_\$ \mathbb{Z}_q^{N+\ell}$ in G_1 based on the LWE assumption. Formally, we have the following claim with proof appeared in Appendix D.1.

Claim 1. $\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \leq Q_{chal} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)$.

Finally, taking all things together, Theorem 3 follows. □

Theorem 4 (wKP Security of mFPRE_1). *Assume that the $\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption holds, then the scheme mFPRE_1 proposed in Fig. 9 has wKP security. More precisely, for any PPT adversary $\mathcal{A}$ that makes at most Q_{rk} queries to $\mathcal{O}_{\text{ReKey}}$ and for any polynomial $\mathfrak{n}$, there exists a PPT algorithm $\mathcal{B}$ against the LWE assumption s.t.* $\mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE}_1,\mathcal{A},\mathfrak{n}}(\lambda) \leq \mathfrak{n} \cdot nQ_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda)$.

Proof of Theorem 4. We prove the theorem via a sequence of games G_0–G_2, where G_0 is the wKP experiment, and in G_2, $\mathcal{A}$ has a negligible advantage.

Game G_0: This is the wKP experiment (cf. Fig. 5). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE}_1,\mathcal{A},\mathfrak{n}}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Let $pk^{(i)} = \mathbf{A}^{(i)}, sk^{(i)} = \mathbf{T}^{(i)}$ denote the public key and secret key of user $i \in [0, \mathfrak{n}]$. In this game, the challenger chooses a random bit $\beta \leftarrow_\$ \{0,1\}$ and answers $\mathcal{A}$'s $\mathcal{O}_{\text{ReKey}}$ queries $(j \in [\mathfrak{n}], f_{\mathbf{M}} \in \mathcal{F}_{\text{lin}})$ as follows:

- If $\beta = 0$, the challenger invokes $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} \leftarrow_\$ \mathsf{FReKGen}(\mathbf{A}^{(0)}, \mathbf{T}^{(0)}, \mathbf{A}^{(j)}, f_{\mathbf{M}})$ and returns $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j}$. More precisely, it samples $\mathbf{S} \leftarrow_\$ \chi^{n\times n}, \mathbf{E} \leftarrow_\$ \chi^{(N+\ell)\times n}$, invokes $\mathbf{R} \leftarrow_\$ \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{A}^{(j)}\mathbf{S}+\mathbf{E}-\binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(0)}, \gamma\big)$, and returns $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$ to $\mathcal{A}$.
- If $\beta = 1$, the challenger invokes $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} \leftarrow_\$ \mathsf{FReKGen}^*(\mathbf{A}^{(0)}, \mathbf{A}^{(j)}, f_{\mathbf{M}})$ which is defined as

$$\mathsf{FReKGen}^*: \ \mathbf{R} \leftarrow_\$ D_{\mathbb{Z}^{(N+\ell)\times N},\gamma} \quad \text{and} \quad \mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right).$$

  Then the challenger returns $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j}$ to the adversary.

Game $\mathsf{G}_{0.t}, t \in [0, \mathfrak{n}]$: It is the same as G_0, except for the reply to $\mathcal{A}$'s $\mathcal{O}_{\text{ReKey}}(j, f_{\mathbf{M}})$ query when $\beta = 0$:

- For $j \leq t$, the challenger uniformly samples $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{(N+\ell)\times n}$ and invokes $\mathbf{R} \leftarrow_\$ \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{U}, \gamma\big)$ to get $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$.
- For $j > t$, the challenger answers the query just like G_0, that is, $\mathbf{R} \leftarrow_\$ \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(0)}, \gamma\big)$ with $\mathbf{S} \leftarrow_\$ \chi^{n\times n}, \mathbf{E} \leftarrow_\$ \chi^{(N+\ell)\times n}$.

Clearly, $\mathsf{G}_{0.0}$ is identical to G_0. Thus, we have $\Pr_0[\mathsf{Win}] = \Pr_{0.0}[\mathsf{Win}]$.

Below we show the computational indistinguishability between $\mathsf{G}_{0.t-1}$ and $\mathsf{G}_{0.t}$ based on the LWE assumption.

Claim 2. For all $t \in [\mathfrak{n}]$, $|\Pr_{0.t-1}[\mathsf{Win}] - \Pr_{0.t}[\mathsf{Win}]| \leq nQ_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)$.

Proof. Firstly, we construct a PPT adversary $\mathcal{B}'$ against the nQ_{rk}-$\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption, such that $|\Pr_{0.t-1}[\mathsf{Win}]-\Pr_{0.t}[\mathsf{Win}]| \le \mathsf{Adv}^{nQ_{rk}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda)$. Then by a standard hybrid argument, we have $\mathsf{Adv}^{nQ_{rk}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda) \le nQ_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)$ and the claim follows.

Algorithm $\mathcal{B}'$. Given a challenge $(\mathbf{A},\mathbf{Z})$, $\mathcal{B}'$ wants to distinguish $\mathbf{Z}=\mathbf{A}\mathbf{S}+\mathbf{E}$ from $\mathbf{Z} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$, where $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times n}$, $\mathbf{S} \leftarrow_{\$} \chi^{n\times nQ_{rk}}$, $\mathbf{E} \leftarrow_{\$} \chi^{(N+\ell)\times nQ_{rk}}$.

$\mathcal{B}'$ is constructed by simulating $\mathsf{G}_{0.t-1}/\mathsf{G}_{0.t}$ for $\mathcal{A}$ as follows. Firstly, $\mathcal{B}'$ sets $pk^{(t)} := \mathbf{A}^{(t)} := \mathbf{A}$ directly for the user t, and invokes KGen honestly to generate $(pk^{(i)}, sk^{(i)})$ for all other users $i \in [0,\mathfrak{n}]\setminus\{t\}$. In particular, $\mathcal{B}'$ owns $sk^{(0)} = \mathbf{T}^{(0)}$. $\mathcal{B}'$ sends $\{pk^{(i)}\}_{i\in[0,\mathfrak{n}]}$ to $\mathcal{A}$. Then $\mathcal{B}'$ chooses a random bit $\beta \leftarrow_{\$} \{0,1\}$ and parses $\mathbf{Z} = (\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_{Q_{rk}}) \in \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$ with each $\mathbf{Z}_k \in \mathbb{Z}_q^{(N+\ell)\times n}$ for $k \in [Q_{chal}]$. On receiving an $\mathcal{O}_{\textsc{ReKey}}(j \in [\mathfrak{n}], f_{\mathbf{M}})$ query from $\mathcal{A}$, if $\beta = 1$, $\mathcal{B}'$ invokes $\mathsf{FReKGen}^*$ to get $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j}$ and returns it to $\mathcal{A}$, the same as $\mathsf{G}_{0.t-1}$ and $\mathsf{G}_{0.t}$. Otherwise, i.e., $\beta = 0$, $\mathcal{B}'$ answers the $\mathcal{O}_{\textsc{ReKey}}(j \in [\mathfrak{n}], f_{\mathbf{M}})$ query in the following way:

- For $j \le t-1$, $\mathcal{B}'$ samples $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times n}$ and invokes $\mathbf{R} \leftarrow_{\$} \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{U}, \gamma\big)$ to get $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} := \left(\mathbf{R} \;\middle|\; \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$, the same as $\mathsf{G}_{0.t-1}$ and $\mathsf{G}_{0.t}$.
- For $j = t$, suppose that this is the k-th $\mathcal{O}_{\textsc{ReKey}}$ query with $k \in [Q_{rk}]$, $\mathcal{B}'$ makes use of $\mathbf{Z}_k$ to invoke $\mathbf{R} \leftarrow_{\$} \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{Z}_k - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(0)}, \gamma\big)$ to get $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to t} := \left(\mathbf{R} \;\middle|\; \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$.

  In the case of $\mathbf{Z} = \mathbf{A}\mathbf{S}+\mathbf{E}$, by parsing $\mathbf{S} = (\mathbf{S}_1 \mid \cdots \mid \mathbf{S}_{Q_{rk}}) \in \mathbb{Z}_q^{n\times nQ_{rk}}$ with each $\mathbf{S}_k \in \mathbb{Z}_q^{n\times n}$ and parsing $\mathbf{E} = (\mathbf{E}_1 \mid \cdots \mid \mathbf{E}_{Q_{rk}}) \in \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$ with each $\mathbf{E}_k \in \mathbb{Z}_q^{(N+\ell)\times n}$, we have $\mathbf{Z}_k = \mathbf{A}\mathbf{S}_k + \mathbf{E}_k = \mathbf{A}^{(t)}\mathbf{S}_k + \mathbf{E}_k$ for $\mathbf{S}_k \leftarrow_{\$} \chi^{n\times n}$ and $\mathbf{E}_k \leftarrow_{\$} \chi^{(N+\ell)\times n}$, and consequently, $\mathcal{B}'$'s simulation is identical to $\mathsf{G}_{0.t-1}$.

  In the case of $\mathbf{Z} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$, we have that $\mathbf{Z}_k$ is uniformly distributed over $\mathbb{Z}_q^{(N+\ell)\times n}$, so $\mathcal{B}'$'s simulation is identical to $\mathsf{G}_{0.t}$.
- For $j > t$, $\mathcal{B}'$ samples $\tilde{\mathbf{S}} \leftarrow_{\$} \chi^{n\times n}$, $\tilde{\mathbf{E}} \leftarrow_{\$} \chi^{(N+\ell)\times n}$ and invokes $\mathbf{R} \leftarrow_{\$} \mathsf{SamplePre}$ $\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{A}^{(j)}\tilde{\mathbf{S}} + \tilde{\mathbf{E}} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(0)}, \gamma\big)$ to get $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to j} := \left(\mathbf{R} \;\middle|\; \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$, the same as $\mathsf{G}_{0.t-1}$ and $\mathsf{G}_{0.t}$.

Finally, $\mathcal{B}'$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}'$ outputs 1 to its own challenger if and only if $\beta' = \beta$.

Now we analyze the advantage of $\mathcal{B}'$. Overall, $\mathcal{B}'$ simulates $\mathsf{G}_{0.t-1}$ for $\mathcal{A}$ in the case $\mathbf{Z} = \mathbf{A}\mathbf{S}+\mathbf{E}$ while simulates $\mathsf{G}_{0.t}$ for $\mathcal{A}$ in the case $\mathbf{Z} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$. Thus $\mathcal{B}'$ successfully distinguishes $\mathbf{Z} = \mathbf{A}\mathbf{S}+\mathbf{E}$ from $\mathbf{Z} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times nQ_{rk}}$ as long as the probability that $\beta' = \beta$ in $\mathsf{G}_{0.t-1}$ differs non-negligibly from that in $\mathsf{G}_{0.t}$. Consequently, we have $\mathsf{Adv}^{nQ_{rk}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda) \ge \big|\Pr_{0.t-1}[\mathsf{Win}] - \Pr_{0.t}[\mathsf{Win}]\big|$, as desired. This completes the proof of Claim 2. ▌

Game G_1: It's the same as G_0, except for the reply to $\mathcal{A}$'s $\mathcal{O}_{\textsc{ReKey}}(j, f_{\mathbf{M}})$ query when $\beta = 0$:

- For all $j \in [\mathfrak{n}]$, the challenger uniformly samples $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times n}$ and uses $\mathbf{U}$ to invoke $\mathbf{R} \leftarrow_{\$} \mathsf{SamplePre}\big(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{U}, \gamma\big)$ to obtain $\mathsf{rk}_{0\to j}^{f_{\mathbf{M}}} := \left(\mathbf{R} \,\middle|\, \begin{smallmatrix}\mathbf{0}\\ \mathbf{M}\end{smallmatrix}\right)$, and returns $\mathsf{rk}_{0\to j}^{f_{\mathbf{M}}}$ to $\mathcal{A}$.

Clearly, $\mathsf{G}_1 = \mathsf{G}_{0.\mathfrak{n}}$ and $\Pr_1[\mathsf{Win}] = \Pr_{0.\mathfrak{n}}[\mathsf{Win}]$. Thus by Claim 2, we have

$$\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \leq \mathfrak{n} \cdot nQ_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda).$$

<u>**Game G_2:**</u> It's the same as G_1, except for the reply to $\mathcal{A}$'s $\mathcal{O}_{\text{ReKey}}(j, f_{\mathbf{M}})$ query when $\beta = 0$. The challenger samples $\mathbf{R}$ by $\mathbf{R} \leftarrow_{\$} D_{\mathbb{Z}^{(N+\ell)\times N},\gamma}$, instead of invoking $\mathbf{R} \leftarrow_{\$} \mathsf{SamplePre}\left(\mathbf{T}^{(0)}, \overline{\mathbf{A}}^{(0)}, \mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times n}, \gamma\right)$ as in G_1.

Since $\gamma \geq O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$, according to the indistinguishability of preimage-sampling of Lemma 3 in Appendix A.2, G_2 is statistically close to G_1. Thus we have $\big|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\big| \leq \mathsf{negl}(\lambda)$.

Finally, note that in G_2, the challenger's reply to $\mathcal{A}$'s $\mathcal{O}_{\text{ReKey}}$ query in the case $\beta = 0$ is identical to that in the case $\beta = 1$. Thus the challenge bit β is completely hidden to $\mathcal{A}$, and we have $\Pr_2[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Theorem 4 follows. □

By plugging Theorem 3 (IND security) and Theorem 4 (wKP security) into Theorem 1 (IND + $\mathsf{wKP} \Rightarrow \mathsf{CPA}$) in Subsect. 3.2, we have the following corollary showing the CPA security of mFPRE_1 based on the LWE assumption.

Corollary 1 (CPA Security of mFPRE_1). *Assume that the $\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption holds, then the scheme mFPRE_1 proposed in Fig. 9 is CPA secure. More precisely, for any PPT adversary $\mathcal{A}$ that makes at most Q_{rk} queries to $\mathcal{O}_{\text{ReKey}}$ and forms a challenge graph G (i.e., subgraph reachable from the vertex of challenge user) in $\mathcal{G}(\mathfrak{n}, \delta, d)$, for any polynomial $\mathfrak{n}$, there exists a PPT algorithm $\mathcal{B}$ against the LWE assumption s.t.*

$$\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE}_1,\mathcal{A},\mathfrak{n}} \leq (2\tau \cdot n\mathfrak{n}Q_{rk} + 2) \cdot \mathfrak{n}^{\sigma+\delta+1} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda),$$

where δ denotes the outdegree, d the depth, τ the pebbling time complexity and σ space complexity for the class $\mathcal{G}(\mathfrak{n}, \delta, d)$, respectively (cf. Appendix B.1).

4.2 The HRA secure Multi-Hop FPRE Scheme mFPRE_2

Parameters. Let $\mathsf{pp}_{\mathsf{LWE}} = (p, q, n, N, L, \ell, \gamma, \varDelta, \chi, \{\chi_v\}_{v\in[0,L]})$ be LWE-related parameters that meet the following conditions:

- $p, q, n, N, L, \ell, \gamma, \varDelta \in \mathbb{N}$ are integers, where $q := p^2$, $\gamma \geq O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$;
- χ is a B-bounded distribution, where B satisfies $\gamma \cdot \omega(\log n) \leq B$.
- For each $v \in [0, L]$, χ_v is the uniform distribution over $[-B_v, B_v]$, where B_v satisfies $B_v \geq 2^{\sqrt[3]{\lambda}} \cdot (nB + NB + \ell\varDelta)B_{v-1}$ for $v \geq 1$ and $B_L \leq \min\{p/4, q/(20N)\}$.

Table 3. Concrete parameters setting, where λ denotes the security parameter and c denotes an arbitrary constant.

Parameters	p	q	n	N	L	ℓ	γ	Δ	B	$B_v\ (v \in [0, L])$
Settings ($L =$ constant)	$2^{\sqrt{\lambda}}$	$2^{2\sqrt{\lambda}}$	λ	λ	c	λ	$\sqrt{\lambda}(\log\lambda)^2$	λ	$\sqrt{\lambda}(\log\lambda)^4$	$(\lambda^2 \cdot 2^{\sqrt[3]{\lambda}+1})^{v+1}$
Settings ($L =$ sub-linear)	$2^{\lambda^{3/4}}$	$2^{2\lambda^{3/4}}$	λ	λ	$c \cdot \sqrt[3]{\lambda}$	λ	$\sqrt{\lambda}(\log\lambda)^2$	λ	$\sqrt{\lambda}(\log\lambda)^4$	$(\lambda^2 \cdot 2^{\sqrt[3]{\lambda}+1})^{v+1}$

More precisely, we describe two settings of parameter in Table 3, one for constant hops ($L = c$) and the other for sub-linear hops ($L = c \cdot \sqrt[3]{\lambda}$), both under sub-exponential modulus q. For simplicity, we assume that all algorithms of our scheme mFPRE_2 take $\mathsf{pp}_{\mathsf{LWE}}$ as an implicit input.

LWE-based Multi-Hop FPRE Scheme mFPRE_2. Our LWE-based FPRE scheme $\mathsf{mFPRE}_2 = (\mathsf{KGen}, \mathsf{FReKGen}, \mathsf{Enc}, \mathsf{FReEnc}, \mathsf{Dec})$ is also for the bounded linear function family $\mathcal{F}_{\text{lin}}$ defined in (9) in Subsect. 4.1, and is shown in Fig. 10.

```
$(pk, sk) \xleftarrow{\$} \mathsf{KGen}$:
    $(\overline{\mathbf{A}} \in \mathbb{Z}_q^{N \times n}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, 1^N)$
    $\underline{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{\ell \times n}$
    $pk := \mathbf{A} = \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}} \in \mathbb{Z}_q^{(N+\ell) \times n}$
    $sk := \mathbf{T}$
    Return $(pk, sk)$

$\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \xleftarrow{\$} \mathsf{FReKGen}(pk^{(i)} = \mathbf{A}^{(i)}, sk^{(i)} = \mathbf{T}^{(i)}, pk^{(j)} = \mathbf{A}^{(j)}, f_{\mathbf{M}} \in \mathcal{F}_{\text{lin}})$:
    $\mathbf{S} \xleftarrow{\$} \chi^{n \times n}$, $\mathbf{E} \xleftarrow{\$} \chi^{(N+\ell) \times n}$
    Parse $\mathbf{A}^{(i)} = \binom{\overline{\mathbf{A}}^{(i)}}{\underline{\mathbf{A}}^{(i)}}$
    $\mathbf{R} \in \mathbb{Z}^{(N+\ell) \times N} \xleftarrow{\$} \mathsf{SamplePre}\left(\mathbf{T}^{(i)}, \overline{\mathbf{A}}^{(i)}, \mathbf{A}^{(j)}\mathbf{S} + \mathbf{E} - \binom{\mathbf{0}}{\mathbf{M}}\underline{\mathbf{A}}^{(i)}, \gamma\right)$
    $\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} := \left(\mathbf{R} \,\middle|\, \begin{matrix}\mathbf{0} \\ \mathbf{M}\end{matrix}\right) \in \mathbb{Z}_p^{(N+\ell) \times (N+\ell)}$    //$\mathbf{M}$ is the description of $f_{\mathbf{M}}$
    Return $\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}}$

$ct_v \xleftarrow{\$} \mathsf{Enc}(pk = \mathbf{A}, \mathbf{m} \in \mathcal{M}, v \in [0, L])$:
    $\mathbf{s} \xleftarrow{\$} \chi_v^n$, $\mathbf{e} \xleftarrow{\$} \chi_v^{N+\ell}$
    $ct_v := \mathbf{A}\mathbf{s} + \mathbf{e} + \binom{\mathbf{0}}{p\mathbf{m}} \in \mathbb{Z}_q^{N+\ell}$

$ct_{v+1}^{(j)} \xleftarrow{\$} \mathsf{FReEnc}(\mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \in \mathbb{Z}_p^{(N+\ell) \times (N+\ell)}$,
        $pk^{(j)} = \mathbf{A}^{(j)}, ct_v^{(i)} \in \mathbb{Z}_q^{N+\ell}, v \in [0, L-1])$:
    $\hat{ct}_{v+1}^{(j)} := \mathsf{rk}_{i \to j}^{f_{\mathbf{M}}} \cdot ct_v^{(i)} \in \mathbb{Z}_q^{N+\ell}$
    $\mathbf{s} \xleftarrow{\$} \chi_{v+1}^n$, $\mathbf{e} \xleftarrow{\$} \chi_{v+1}^{N+\ell}$
    $ct_{v+1}^{(j)} := \hat{ct}_v^{(j)} + \mathbf{A}^{(j)}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^{N+\ell}$
    Return $ct_{v+1}^{(j)}$

$\mathbf{m} \leftarrow \mathsf{Dec}(sk = \mathbf{T}, ct \in \mathbb{Z}_q^{N+\ell})$:
    Parse $ct = \binom{\overline{ct} \in \mathbb{Z}_q^N}{\underline{ct} \in \mathbb{Z}_q^\ell}$
    $(\mathbf{s}, \overline{\mathbf{e}}) \xleftarrow{\$} \mathsf{Invert}(\mathbf{T}, \overline{ct})$
    $\tilde{\mathbf{m}} = (\tilde{m}_1, \ldots, \tilde{m}_\ell) := \underline{ct} - \underline{\mathbf{A}}\mathbf{s}$
    For $i \in [\ell]$ :  $m_i := \lceil \tilde{m}_i / p \rfloor$
    Return $\mathbf{m} = (m_1, m_2, \ldots, m_\ell)$
```

Fig. 10. The LWE-based Multi-Hop FPRE scheme mFPRE_2 for $\mathcal{F}_{\text{lin}}$. For ease of reading, we emphasize different parts with the CPA secure scheme mFPRE_1 in gray boxes .

The analysis for the correctness and fine-grained L-hop correctness of mFPRE_2 are similar to those for mFPRE_1, and we put the formal analysis in Appendix C.

Next, we show the IND security, wKP security and SH security of mFPRE_2 via the following three theorems. Then together with Theorem 2 (IND + wKP + $\mathsf{SH} \Rightarrow \mathsf{HRA}$) in Subsect. 3.2, it yields the HRA security of our scheme mFPRE_2.

Theorem 5 (IND Security of mFPRE_2). *Assume that the $\mathsf{LWE}_{n,q,\chi_i,N+\ell}$-assumption holds for all $i \in [0, L]$, then the scheme mFPRE_2 proposed in Fig. 10 has IND security. More precisely, for any PPT adversary $\mathcal{A}$ that make at most Q_{chal} queries to $\mathcal{O}_{\text{CHAL}}$, there exist PPT algorithms $\mathcal{B}_0, \ldots, \mathcal{B}_L$ against the LWE assumptions such that $\mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE}_2,\mathcal{A}}(\lambda) \leq Q_{chal} \cdot \sum_{i=0}^{L} \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi_i,N+\ell],\mathcal{B}_i}(\lambda)$.*

We postpone the proof of Theorem 5 to Appendix D.2.

Theorem 6 (wKP **Security of** mFPRE_2)**.** *Assume that the* $\mathsf{LWE}_{n,q,\chi,N+\ell}$*-assumption holds, then the scheme* mFPRE_2 *proposed in Fig. 10 has* wKP *security. More precisely, for any PPT adversary* $\mathcal{A}$ *that makes at most* Q_{rk} *queries to* $\mathcal{O}_{\text{ReKey}}$ *and for any polynomial* $\mathfrak{n}$*, there exists a PPT algorithm* $\mathcal{B}$ *against the LWE assumption s.t.* $\mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE}_2,\mathcal{A},\mathfrak{n}}(\lambda) \le \mathfrak{n} \cdot nQ_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda)$.

Note that the KGen and $\mathsf{FReKGen}$ algorithms of scheme mFPRE_2 are the same as those of mFPRE_1 in Subsect. 4.1, so does the wKP security. Consequently, the proof of Theorem 6 is identical to that for Theorem 4 and we omit it.

Theorem 7 (SH **Security of** mFPRE_2)**.** *The scheme* mFPRE_2 *proposed in Fig. 10 has* SH *security. More precisely, for any (unbounded) adversary* $\mathcal{A}$*, we have* $\mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE}_2,\mathcal{A}}(\lambda) \le \mathsf{negl}(\lambda)$.

Proof of Theorem 7. To show that $\mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE}_2,\mathcal{A}}(\lambda) \le \mathsf{negl}(\lambda)$, we first elaborate the SH experiment $\mathsf{Exp}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{A}}$ defined in Fig. 6.

Let $pk^{(i)} = \mathbf{A}^{(i)}, sk^{(i)} = \mathbf{T}^{(i)}$ denote the public key and secret key of user $i \in \{0,1\}$. In the experiment, the challenger initiates $\mathcal{Q}_f := \perp$, $\mathcal{L} := \perp$, $\mathsf{ctr} := 0$, chooses $\beta \leftarrow_{\$} \{0,1\}$ and answers $\mathcal{A}$'s $\mathcal{O}_{\text{ReKey}}, \mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Chal}}$ queries as follows:

- On receiving an $\mathcal{O}_{\text{ReKey}}(f_{\mathbf{M}})$ query from $\mathcal{A}$, the challenger adds $f_{\mathbf{M}}$ to $\mathcal{Q}_f$, invokes $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to1} \leftarrow_{\$} \mathsf{FReKGen}(\mathbf{A}^{(0)}, \mathbf{T}^{(0)}, \mathbf{A}^{(1)}, f_{\mathbf{M}})$, and returns $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to1}$ to $\mathcal{A}$.
- On receiving an $\mathcal{O}_{\text{Enc}}(\mathbf{m}, v)$ query from $\mathcal{A}$, the challenger increases the counter ctr, invokes $ct^{(0)}_v \leftarrow_{\$} \mathsf{Enc}(\mathbf{A}^{(0)}, \mathbf{m}, v)$, adds the tuple $(\mathsf{ctr}, (ct^{(0)}_v, v))$ to $\mathcal{L}$, and returns the ciphertext $ct^{(0)}_v$ along with the counter ctr to $\mathcal{A}$.
- On receiving an $\mathcal{O}_{\text{Chal}}(k, f_{\mathbf{M}})$ query from $\mathcal{A}$, the challenger first retrieves $(k, (ct^{(0)}_v, v))$ from $\mathcal{L}$ by the counter k, and returns $\perp$ to $\mathcal{A}$ directly if the retrieval fails. Otherwise, the challenger answers the query as follows.
  - If $\beta = 0$, the challenger first generates $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to1} \leftarrow_{\$} \mathsf{FReKGen}(\mathbf{A}^{(0)}, \mathbf{T}^{(0)}, \mathbf{A}^{(1)}, f_{\mathbf{M}})$ in the case $f_{\mathbf{M}} \notin \mathcal{Q}_f$ (i.e., $\mathsf{rk}^{f_{\mathbf{M}}}_{0\to1}$ has not been generated yet), then it computes the re-encryption $ct^{(1)}_{v+1} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f_{\mathbf{M}}}_{0\to1}, ct^{(0)}_v, v)$, i.e.,
  $$ct^{(1)}_{v+1} := \hat{ct}^{(1)}_{v+1} + \mathbf{A}^{(1)}\mathbf{s}' + \mathbf{e}' \text{ with } \hat{ct}^{(1)}_{v+1} := \mathsf{rk}^{f_{\mathbf{M}}}_{0\to1} \cdot ct^{(0)}_v, \mathbf{s}' \leftarrow_{\$} \chi^n_{v+1}, \mathbf{e}' \leftarrow_{\$} \chi^{N+\ell}_{v+1}, \tag{11}$$
  and returns the re-encrypted ciphertext $ct^{(1)}_{v+1}$ to $\mathcal{A}$.
  - If $\beta = 1$, the challenger generates $ct^{(1)}_{v+1} \leftarrow_{\$} \mathsf{Enc}(\mathbf{A}^{(1)}, f_{\mathbf{M}}(\mathbf{m}) = \mathbf{Mm}, v+1)$ freshly, i.e.,
  $$ct^{(1)}_{v+1} := \mathbf{A}^{(1)}\mathbf{s} + \mathbf{e} + \binom{\mathbf{0}}{p\mathbf{Mm}} \text{ with } \mathbf{s} \leftarrow_{\$} \chi^n_{v+1}, \mathbf{e} \leftarrow_{\$} \chi^{N+\ell}_{v+1}, \tag{12}$$
  and returns the fresh ciphertext $ct^{(1)}_{v+1}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs β', and the advantage of $\mathcal{A}$ is defined by $\mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE}_2,\mathcal{A}}(\lambda) = |\Pr[\beta' = \beta] - \frac{1}{2}| = \frac{1}{2} \cdot \big|\Pr[\beta' = 1 \,|\, \beta = 0] - \Pr[\beta' = 1 \,|\, \beta = 1]\big|$.

Below we analyze $\mathcal{A}$'s advantage. We will show that $\mathcal{A}$ has negligible advantage in distinguishing $\beta = 0$ and $\beta = 1$. More precisely, we note that the only differences between $\beta = 0$ and $\beta = 1$ are the replies $ct_{v+1}^{(1)}$ for $\mathcal{O}_{\textsc{Chal}}(k, f_{\mathbf{M}})$ queries, and we will show that the distributions of $ct_{v+1}^{(1)}$ in (11) are statistically close to the distributions of $ct_{v+1}^{(1)}$ in (12), from the point of view of $\mathcal{A}$.

In the case of $\beta = 0$, $ct_{v+1}^{(1)}$ is generated according to (11), and by a similar analysis as that for **Fine-Grained L-Hop Correctness**, we have

$$\hat{ct}_{v+1}^{(1)} = \mathbf{A}^{(1)} \underbrace{\mathbf{S}\mathbf{s}_0}_{:=\mathbf{s}_1} + \underbrace{\mathbf{E}\mathbf{s}_0 + \mathbf{R}_1\overline{\mathbf{e}_0} + \binom{\mathbf{0}}{\mathbf{M}\underline{\mathbf{e}_0}}}_{:=\mathbf{e}_1} + \binom{\mathbf{0}}{p\mathbf{M}\mathbf{m}},$$

where $\mathbf{s}_0 \leftarrow_{\$} \chi_v^n$, $\mathbf{e}_0 = \binom{\overline{\mathbf{e}_0}}{\underline{\mathbf{e}_0}} \leftarrow_{\$} \chi_v^{N+\ell}$, $\mathbf{S} \leftarrow_{\$} \chi^{n\times n}$, $\mathbf{E} \leftarrow_{\$} \chi^{(N+\ell)\times n}$, and it follows that $\|\mathbf{s}_1\|_\infty \le nBB_v$ and $\|\mathbf{e}_1\|_\infty \le (nB + NB + \ell\Delta)B_v$. Then for $ct_{v+1}^{(1)}$ in (11), it holds that

$$ct_{v+1}^{(1)} = \hat{ct}_{v+1}^{(1)} + \mathbf{A}^{(1)}\mathbf{s}' + \mathbf{e}' = \mathbf{A}^{(1)}(\mathbf{s}_1 + \mathbf{s}') + (\mathbf{e}_1 + \mathbf{e}') + \binom{\mathbf{0}}{p\mathbf{M}\mathbf{m}}, \tag{13}$$

where $\mathbf{s}' \leftarrow_{\$} \chi_{v+1}^n, \mathbf{e}' \leftarrow_{\$} \chi_{v+1}^{N+\ell}$. Since $\chi_{v+1} = [-B_{v+1}, B_{v+1}]$ with $B_{v+1} \ge 2^{\sqrt[3]{\lambda}} \cdot (nB + NB + \ell\Delta)B_v \ge 2^{\sqrt[3]{\lambda}} \cdot \{\|\mathbf{s}_1\|_\infty, \|\mathbf{e}_1\|_\infty\}$, by Lemma 5 (the Smudging Lemma), $\mathbf{s}'$ and $\mathbf{e}'$ smudge $\mathbf{s}_1$ and $\mathbf{e}_1$ respectively, so that the $ct_{v+1}^{(1)}$ in (13) is statistically close to the $ct_{v+1}^{(1)}$ in (12) with statistical distance at most $(n + N + \ell)/2^{\sqrt[3]{\lambda}}$.

By a union bound over all $\mathcal{O}_{\textsc{Chal}}$ queries made by $\mathcal{A}$ (say Q_{chal} number of queries), all replies $ct_{v+1}^{(1)}$ in (13) (corresponding to $\beta = 0$) are statistically close to those in (12) (corresponding to $\beta = 1$) with statistical distance at most $Q_{chal} \cdot (n + N + \ell)/2^{\sqrt[3]{\lambda}} = \mathsf{negl}(\lambda)$. Consequently, $\mathcal{A}$ has negligible advantage in distinguishing $\beta = 0$ and $\beta = 1$, and Theorem 7 follows. □

By plugging Theorem 5 (IND security), Theorem 6 (wKP security) and Theorem 7 (SH security) into Theorem 2 ($\mathsf{IND} + \mathsf{wKP} + \mathsf{SH} \Rightarrow \mathsf{HRA}$) in Subsect. 3.2, we have the following corollary showing the HRA security of our scheme mFPRE_2 based on the LWE assumption.

Corollary 2 (HRA Security of mFPRE_2). *Assume that the $\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption and the $\mathsf{LWE}_{n,q,\chi_i,N+\ell}$-assumption hold for all $i \in [0, L]$, then the scheme mFPRE_2 proposed in Fig. 10 is HRA secure. More precisely, for any PPT adversary $\mathcal{A}$ that makes at most Q_{rk} queries to $\mathcal{O}_{\textsc{ReKey}}$ and forms a challenge graph G (i.e., subgraph reachable from the vertex of challenge user) in $\mathcal{G}(\mathfrak{n}, \delta, d)$, for any polynomial $\mathfrak{n}$, there exists PPT algorithms $\mathcal{B}_0, \ldots, \mathcal{B}_L$ and $\mathcal{B}$ against the LWE assumption s.t.*

$$\mathsf{Adv}^{\mathsf{HRA}}_{\mathsf{mFPRE}_2,\mathcal{A},\mathfrak{n}} \le \Big(2\textstyle\sum_{i=0}^{L} \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi_i,N+\ell],\mathcal{B}_i}(\lambda) + 2\tau \cdot n\mathfrak{n}Q_{rk} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)\Big) \cdot \mathfrak{n}^{\sigma+\delta+1} + \mathsf{negl}(\lambda),$$

where δ denotes the outdegree, d the depth, τ the pebbling time complexity and σ space complexity for the class $\mathcal{G}(\mathfrak{n}, \delta, d)$, respectively (cf. Appendix B.1).

Acknowledgments. We would like to thank the reviewers for their valuable comments and helpful suggestions. Yunxiao Zhou, Shengli Liu and Shuai Han were partially supported by the National Key R&D Program of China under Grant 2022YFB2701500, National Natural Science Foundation of China (Grant Nos. 61925207, 62372292), Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), and Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20200185).

References

[1] Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC. pp. 99–108. ACM Press (May 1996)

[2] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (Aug 2009)

[3] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (Apr 2012)

[4] Ateniese, G., Benson, K., Hohenberger, S.: Key-private proxy re-encryption. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 279–294. Springer, Heidelberg (Apr 2009)

[5] Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. In: NDSS 2005. The Internet Society (Feb 2005)

[6] Bennett, C.H.: Time/space trade-offs for reversible computation. SIAM J. Comput. 18(4), 766–776 (1989), https://doi.org/10.1137/0218053

[7] Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (May / Jun 1998)

[8] Chandran, N., Chase, M., Liu, F.H., Nishimaki, R., Xagawa, K.: Re-encryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 95–112. Springer, Heidelberg (Mar 2014)

[9] Cohen, A.: What about bob? The inadequacy of CPA security for proxy reencryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 287–316. Springer, Heidelberg (Apr 2019)

[10] Davidson, A., Deo, A., Lee, E., Martin, K.: Strong post-compromise secure proxy re-encryption. In: Jang-Jaccard, J., Guo, F. (eds.) ACISP 19. LNCS, vol. 11547, pp. 58–77. Springer, Heidelberg (Jul 2019)

[11] Fan, X., Liu, F.H.: Proxy re-encryption and re-signatures from lattices. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 19. LNCS, vol. 11464, pp. 363–382. Springer, Heidelberg (Jun 2019)

[12] Fuchsbauer, G., Kamath, C., Klein, K., Pietrzak, K.: Adaptively secure proxy re-encryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 317–346. Springer, Heidelberg (Apr 2019)

[13] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009)

[14] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008)

[15] Jafargholi, Z., Kamath, C., Klein, K., Komargodski, I., Pietrzak, K., Wichs, D.: Be adaptive, avoid overcommitting. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 133–163. Springer, Heidelberg (Aug 2017)

[16] Lai, J., Huang, Z., Au, M.H., Mao, X.: Constant-size CCA-secure multi-hop unidirectional proxy re-encryption from indistinguishability obfuscation. In: Susilo, W., Yang, G. (eds.) ACISP 18. LNCS, vol. 10946, pp. 805–812. Springer, Heidelberg (Jul 2018)

[17] Lai, J., Huang, Z., Au, M.H., Mao, X.: Constant-size cca-secure multi-hop unidirectional proxy re-encryption from indistinguishability obfuscation. Theoretical Computer Science 847, 1–16 (2020), https://www.sciencedirect.com/science/article/pii/S0304397520305302

[18] Miao, P., Patranabis, S., Watson, G.J.: Unidirectional updatable encryption and proxy re-encryption from DDH. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part II. LNCS, vol. 13941, pp. 368–398. Springer (2023)

[19] Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (Aug 2011)

[20] Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (Apr 2012)

[21] Phong, L.T., Wang, L., Aono, Y., Nguyen, M.H., Boyen, X.: Proxy re-encryption schemes with key privacy from LWE. Cryptology ePrint Archive, Report 2016/327 (2016), https://eprint.iacr.org/2016/327

[22] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005)

[23] Smith, T.: Dvd jon: Buy drm-less tracks from apple itunes (2005), https://www.theregister.com/2005/03/18/itunes_pymusique/

[24] Zhou, Y., Liu, S., Han, S., Zhang, H.: Fine-grained proxy re-encryption: Definitions & constructions from LWE. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VI. LNCS, vol. 14443, pp. 199–231. Springer (2023)

Supplementary Material

A Additional Preliminaries

A.1 Multi-Hop Proxy Re-Encryption

We recall the syntax of multi-hop proxy re-encryption according to [12].

Definition 9 (Multi-Hop PRE). *A multi-hop proxy re-encryption (mPRE) scheme is associated with a message space $\mathcal{M}$, a maximum level $L \in \mathbb{N}$ and defined with five PPT algorithms* $\mathsf{mPRE} = (\mathsf{KGen}, \mathsf{ReKGen}, \mathsf{Enc}, \mathsf{ReEnc}, \mathsf{Dec})$.

- $(pk, sk) \leftarrow_{\$} \mathsf{KGen}$*: The key generation algorithm outputs a pair of public key and secret key* (pk, sk).
- $\mathsf{rk}_{i\to j} \leftarrow_{\$} \mathsf{ReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)})$*: Taking as input a public-secret key pair* $(pk^{(i)}, sk^{(i)})$ *and another public key* $pk^{(j)}$*, the re-encryption key generation algorithm outputs a re-encryption key* $\mathsf{rk}_{i\to j}$ *that allows re-encrypting ciphertexts intended to* i *into ciphertexts encrypted for* j.
- $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, m, v)$*: Taking as input* pk*, a message* $m \in \mathcal{M}$ *and a level* $v \in [0, L]$*, the encryption algorithm outputs a* v*-level ciphertext* ct_v.
- $ct^{(j)}_{v+1} \leftarrow_{\$} \mathsf{ReEnc}(\mathsf{rk}_{i\to j}, ct^{(i)}_v, v)$*: Taking as input a re-encryption key* $\mathsf{rk}_{i\to j}$ *and a ciphertext* $ct^{(i)}_v$ *intended for* i *and its level* $v \in [0, L-1]$*, the re-encryption algorithm outputs a* $(v+1)$*-level ciphertext* $ct^{(j)}_{v+1}$ *re-encrypted for* j*. We denote it by* $ct^{(i)}_v \xrightarrow{\mathsf{rk}_{i\to j}} ct^{(j)}_{v+1}$.
- $m \leftarrow \mathsf{Dec}(sk, ct)$*: Taking as input a secret key* sk *and a ciphertext* ct*, the deterministic decryption algorithm outputs a message* m.

Correctness. *For all* $m \in \mathcal{M}, v \in [0, L]$, $(pk, sk) \leftarrow_{\$} \mathsf{KGen}$, $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, m, v)$, *it holds that* $\mathsf{Dec}(sk, ct_v) = m$.
L***-Hop Correctness.*** *For all* $m \in \mathcal{M}$*, user indices* $i_0, i_1, \cdots, i_L$, $(pk^{(i_j)}, sk^{(i_j)}) \leftarrow_{\$} \mathsf{KGen}$ *with* $j \in [0, L]$*,* 0*-level ciphertext* $ct^{(i_0)}_0 \leftarrow_{\$} \mathsf{Enc}(pk^{(i_0)}, m, 0)$ *and re-encryption hops* $ct^{(i_0)}_0 \xrightarrow{\mathsf{rk}_{i_0\to i_1}} ct^{(i_1)}_1 \xrightarrow{\mathsf{rk}_{i_1\to i_2}} \cdots \xrightarrow{\mathsf{rk}_{i_{L-1}\to i_L}} ct^{(i_L)}_L$*, where each* $\mathsf{rk}_{i_{j-1}\to i_j} \leftarrow_{\$} \mathsf{ReKGen}(pk^{(i_{j-1})}, sk^{(i_{j-1})}, pk^{(i_j)})$ *and each* $ct^{(i_j)}_j \leftarrow_{\$} \mathsf{ReEnc}(\mathsf{rk}_{i_{j-1}\to i_j}, ct^{(i_{j-1})}_{j-1}, j-1)$*, it holds that for all* $j \in [L]$, $\mathsf{Dec}(sk^{(i_j)}, ct^{(i_j)}_j) = m$.

Note that the above mPRE is defined as a *non-interactive* one, since $sk^{(j)}$ is not needed in algorithm ReKGen for the generation of $\mathsf{rk}_{i\to j}$.

A.2 Lattice Backgrounds

Definition 10 (Discrete Gaussian Distribution). *The Gaussian function with parameter* s *and center* $\mathbf{c} \in \mathbb{R}^n$ *is defined as* $\rho_{s,\mathbf{c}} : \mathbb{R}^n \to \mathbb{R}$, $\rho_{s,\mathbf{c}}(\mathbf{x}) := e^{-\pi\|\mathbf{x}-\mathbf{c}\|^2/s^2}$*. For a countable set* $\mathcal{S} \subset \mathbb{R}^n$*, the discrete Gaussian distribution* $D_{\mathcal{S},s,\mathbf{c}}$ *parameterized with* s *and* $\mathbf{c}$ *is defined as* $D_{\mathcal{S},s,\mathbf{c}}(\mathbf{x}) := \rho_{s,\mathbf{c}}(\mathbf{x}) / \sum_{\mathbf{x}\in\mathcal{S}} \rho_{s,\mathbf{c}}(\mathbf{x})$ *for* $\mathbf{x} \in \mathcal{S}$ *and* $D_{\mathcal{S},s,\mathbf{c}}(\mathbf{x}) := 0$ *for* $\mathbf{x} \notin \mathcal{S}$*. Usually,* s *is omitted when* $s = 1$ *and* $\mathbf{c}$ *is omitted if* $\mathbf{c} = \mathbf{0}$.

Below we recall the LWE and multi-secret LWE assumptions, where both the secret vector and the error vector are sampled from the same distribution (say χ). This version of LWE was formalized by Applebaum et al. [2] and was proved at least as hard as the usual definition of LWE where the secret vector is sampled uniformly at random.

Definition 11 (LWE Assumption [22, 2]). *Let $n, m, q \in \mathbb{N}$ and χ be a distribution over $\mathbb{Z}_q$. The $\mathsf{LWE}_{n,q,\chi,m}$-assumption requires that for any PPT adversary $\mathcal{A}$, it's advantage function satisfies $\mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,m],\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{As}+\mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1]\big| \le \mathsf{negl}(\lambda)$, where $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \leftarrow_\$ \chi^n$, $\mathbf{e} \leftarrow_\$ \chi^m$, $\mathbf{u} \leftarrow_\$ \mathbb{Z}_q^m$.*

For $Q \in \mathbb{N}$, the Q-$\mathsf{LWE}_{n,q,\chi,m}$-assumption requires that for any PPT $\mathcal{A}$, its advantage satisfies $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,\chi,m],\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{AS}+\mathbf{E}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{U}) = 1]\big| \le \mathsf{negl}(\lambda)$, where $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$, $\mathbf{S} \leftarrow_\$ \chi^{n\times Q}$, $\mathbf{E} \leftarrow_\$ \chi^{m\times Q}$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times Q}$.

A simple hybrid argument shows that $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,\chi,m]}(\lambda) \le Q \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,m]}(\lambda)$.

In [1, 20], an algorithm named $\mathsf{TrapGen}$ is proposed to sample a "nearly" uniform random matrix $\mathbf{A}$ along with a low-norm trapdoor matrix $\mathbf{T_A}$ such that $\mathbf{T_A} \cdot \mathbf{A} = \mathbf{0}$ (cf. Lemma 1). Meanwhile, another algorithm called Invert is proposed to make use of $\mathbf{T_A}$ to invert an LWE sample $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ to obtain $\mathbf{s}$ and $\mathbf{e}$ (cf. Lemma 2).

Lemma 1 ([1, 20]). *There exists a PPT algorithm $\mathsf{TrapGen}$ that takes as input positive integers n, q ($q \ge 2$) and a sufficiently large $m = O(n \log q)$, outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and a trapdoor matrix $\mathbf{T_A} \in \mathbb{Z}_q^{m\times m}$ such that $\mathbf{A}$ is statistically close to the uniform distribution, $\mathbf{T_A} \cdot \mathbf{A} = \mathbf{0}$, and $\|\widetilde{\mathbf{T}}_\mathbf{A}\| \le O(\sqrt{n \log q})$, where $\widetilde{\mathbf{T}}_\mathbf{A}$ denotes the Gram-Schmidt orthogonalization of $\mathbf{T_A}$.*

Lemma 2 ([20, Theorem 5.4]). *There exists a deterministic polynomial-time algorithm Invert that takes as inputs the trapdoor information $\mathbf{T_A}$ and a vector $\mathbf{v} := \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$ with $\mathbf{s} \in \mathbb{Z}_q^n$ and $\|\mathbf{e}\| \le q/(10\sqrt{m})$, and outputs $\mathbf{s}$ and $\mathbf{e}$.*

Lemma 3 ([14]). *Let $n, m, q \in \mathbb{N}$ with $q \ge 2$, and $\gamma \ge O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$.*

- **Preimage-sampling.** *Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be a matrix with a trapdoor $\mathbf{T_A}$. Let $\mathbf{B} \in \mathbb{Z}_q^{m'\times n}$. There exists a PPT algorithm $\mathsf{SamplePre}(\mathbf{T_A}, \mathbf{A}, \mathbf{B}, \gamma)$ that outputs a matrix $\mathbf{R} \in \mathbb{Z}^{m'\times m}$ which is sampled from a distribution statistically close to $D_{\Lambda_q^{\mathbf{B}}(\mathbf{A}),\gamma}$ and satisfies $\mathbf{R} \cdot \mathbf{A} = \mathbf{B}$ and $\|\mathbf{R}\|_\infty \le \gamma \cdot \omega(\log n)$ (except with a negligible probability).*
- **Indistinguishability of preimage-sampling.** *Let $\mathsf{TrapGen}$ be the algorithm defined in Lemma 1. Let $m \ge O(n \log q)$. Then we have $(\mathbf{A}, \mathbf{R}, \mathbf{B}) \approx_s (\mathbf{A}, \mathbf{R}', \mathbf{B}')$, where the probability is over $(\mathbf{A}, \mathbf{T_A}) \leftarrow_\$ \mathsf{TrapGen}(n, q, m)$, $\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{m'\times m}$, $\mathbf{R} \leftarrow_\$ \mathsf{SamplePre}(\mathbf{T_A}, \mathbf{A}, \mathbf{B}, \gamma)$, $\mathbf{R}' \leftarrow_\$ D_{\mathbb{Z}^{m'\times m},\gamma}$, and $\mathbf{B}' := \mathbf{R}' \cdot \mathbf{A}$.*

Lemma 4 (Randomness Extraction, Particular case of [19, Lemma 2.3]). *Let $n, m, q \in \mathbb{N}$, $\epsilon \in (0, 1)$. Suppose that $\mathbf{r}$ is chosen from some distribution over $\mathbb{Z}_q^m$ s.t. for q's prime factor p it holds that $\mathbf{H}_\infty(\mathbf{r} \bmod p) \ge 2n \log q + 2\log(\frac{1}{\epsilon})$. Then for $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$, $\mathbf{u} \leftarrow_\$ \mathbb{Z}_q^n$, we have $\Delta\big((\mathbf{A}, \mathbf{r}^\top \cdot \mathbf{A}), (\mathbf{A}, \mathbf{u}^\top)\big) \le \epsilon$.*

Lemma 5 (Smudging Lemma, [3, Lemma 1]). *Let B, B' be positive integers, and $e \in [-B, B]$ a fixed integer. Then for a uniformly chosen $e' \leftarrow_{\$} [-B', B']$, it holds that $\Delta(e + e', e') = B/B'$.*

B Proofs of Theorem 1 and Theorem 2

In this section, we will provide the proofs of Theorem 1 and Theorem 2. Our proofs mainly follow the frameworks of [15, 12], so we will first recall their frameworks in Appendix B.1. Then we present the proofs of Theorem 1 and Theorem 2 in Appendix B.2 and Appendix B.3, respectively.

B.1 Additional Preliminaries

Notations. For two sets $\mathcal{X}, \mathcal{Y}$ we write $\mathcal{X} \Delta \mathcal{Y}$ for the symmetric difference. With $\mathsf{X} \equiv \mathsf{Y}$ we denote that algorithm X has exactly the same input/output distribution as Y. For graphs, let $G = (\mathcal{V}, \mathcal{E})$ denote a directed graph with vertices $\mathcal{V}$ (usually $\mathcal{V} = [\mathfrak{n}]$ for some $\mathfrak{n} \in \mathbb{N}$) and edges $\mathcal{E} \subseteq \mathcal{V}^2$. The indegree (resp., outdegree) of a vertex is defined as the number of edges coming in to (resp., going out of) that vertex. The indegree (resp., outdegree) of the graph is the maximum indegree (resp., outdegree) over all the vertices. A vertex with indegree (resp., outdegree) zero is called a source (resp., sink). Let $\mathcal{G}(\mathfrak{n}, \delta, d)$ denote the class of all directed graphs with $\mathfrak{n}$ vertices, outdegree δ and depth d. A vertex i is connected to another vertex j (or alternatively j is reachable from i) if there is a directed path from i to j in G. $\mathsf{children}(i, G)$ refers to the set of vertices j such that $(i, j) \in \mathcal{E}$. “DAG” abbreviates directed acyclic graph.

B.1.1 Pebbling Game

The classical reversible black pebbling game on DAGs was introduced in [6] to model reversible computation. In [12], Fuchsbauer et al. defined a variant in order to adapt this technique for application to PREs. We adopt their pebbling rule: a pebble can be placed on or removed from a vertex i if all its children $\mathsf{children}(i, G)$ carry a pebble.

Definition 12 (Pebbling Game [12]). *A reversible pebbling of a directed acyclic graph $G = (\mathcal{V}, \mathcal{E})$ with a unique source vertex i^* is a sequence $\mathcal{P} := (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ of pebbling configurations $\mathcal{P}_t \subseteq \mathcal{V}$ with $t \in [0, \tau]$. Two subsequent configurations differ only in one vertex and the following rule is respected in a move: a pebble can be placed on or removed from a vertex iff all its children carry a pebble. That is, $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ is a valid sequence iff*

$$\forall t \in [\tau]\ \exists! i \in \mathcal{P}_{t-1} \Delta \mathcal{P}_t \quad and \quad \mathsf{children}(i, G) \subseteq \mathcal{P}_{t-1}.$$

Starting with an empty graph (i.e., $\mathcal{P}_0 = \emptyset$), the goal of the game is to place a pebble on the source (i.e., $i^* \in \mathcal{P}_\tau$).

For a DAG G, let $\mathcal{P}_G$ denote the set of all valid reversible pebbling sequences (as per Def. 12) for G. The *time complexity* of a *particular* sequence $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ for a DAG G is defined as $\tau_G(\mathcal{P}) := \tau$, whereas its *space complexity* is defined as $\sigma_G(\mathcal{P}) := \max_{t \in [0,\tau]} |\mathcal{P}_t|$.

Definition 13 (Time- and space-complexity of a class of DAGs [12]). *We say that a class of DAGs $\mathcal{G}$ has time complexity τ and space complexity σ if*

$$\forall G \in \mathcal{G}, \ \exists \mathcal{P} \in \mathcal{P}_G : \tau_G(\mathcal{P}) \le \tau \quad and \quad \sigma_G(\mathcal{P}) \le \sigma.$$

In [12], Fuchsbauer et al. analysed time complexity and space complexity of different DAGs, including arbitrary DAGs, complete binary trees and chains. Thus, we have following lemma:

Lemma 6 (Concrete Bounds on Pebbling Time Complexity σ and Space Complexity τ [12]). *An arbitrary graph class $\mathcal{G}(\mathfrak{n}, \delta, d)$ has time-complexity $\tau = (2\delta)^d$ and space-complexity $\sigma = (\delta+1) \cdot d$.*

Complete binary trees of size $\mathfrak{n}$, i.e., $\mathcal{B}(\mathfrak{n}) = \mathcal{G}(\mathfrak{n}, 2, \log \mathfrak{n})$, have time-complexity $\tau = \mathfrak{n}^2$ and space-complexity $\sigma = 3 \cdot \log \mathfrak{n}$.

Chains of length $\mathfrak{n}$, i.e., $\mathcal{C}(\mathfrak{n}) = \mathcal{G}(\mathfrak{n}, 1, \mathfrak{n})$, have time-complexity $\tau = 3^{\log \mathfrak{n}}$ and space-complexity $\sigma = \log \mathfrak{n} + 1$.

B.1.2 Framework of Jafargholi et al. [15]

Jafargholi et al. [15] proposed a framework that can help us reduce the security loss of reduction when we try to raise selective security to adaptive security. Fuchsbauer et al. [12] applied this framework on the proof of adaptive security of PRE schemes. Below we recall some useful definitions and theorems from [15].

We consider a game described via a challenger G which interacts with an adversary $\mathcal{A}$. At the end of the game, G outputs a decision bit b and we let $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{G}}$ denote the advantage of $\mathcal{A}$ against G.

Let $\mathcal{W}$ denote the set of information that the adversary $\mathcal{A}$ initially has to commit. The selectivized game is defined as follows.

Definition 14 (Selectivized Game [15]). *Given an (adaptive) game G and some function $g : \{0,1\}^* \to \mathcal{W}$, the selectivized game $\mathsf{H} = \mathsf{SEL}_{\mathcal{W}}[\mathsf{G}, g]$ is defined as follows. The adversary $\mathcal{A}$ first sends a commitment $w \in \mathcal{W}$ to H. Then H runs the challenger G against $\mathcal{A}$, at the end of which G outputs a bit b'. Let $\mathsf{transcript}$ denote all communication exchanged between G and $\mathcal{A}$. If $g(\mathsf{transcript}) = w$, then H outputs the bit b' and else it outputs 0.*

Note that the selectivized game gets a commitment w from the adversary $\mathcal{A}$ but essentially ignores it during the rest of the game. Only, at the very end of the game, it checks that the commitment matches what actually happened during the game.

Let $\mathcal{U}$ denote the set of partial information which is associated to $\mathcal{A}$'s partial choices and assume $|\mathcal{U}| \ll |\mathcal{W}|$. The further selectivized game is defined as follows.

Definition 15 (Further Selectivized Game [15]). *Assume $\hat{\mathsf{H}}$ is a (partially selective) game which expects to receive some commitment $u \in \mathcal{U}$ from the adversary in the first round. Given functions $g : \{0,1\}^* \to \mathcal{W}$ and $h : \mathcal{W} \to \mathcal{U}$, the further selectivized game $\mathsf{H} = \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}, g, h]$ is defined as follows. The adversary $\mathcal{A}$ first sends a commitment $w \in \mathcal{W}$ to H and H begins running $\hat{\mathsf{H}}$ and passes it $u = h(w)$. It then continues running the game between $\hat{\mathsf{H}}$ and $\mathcal{A}$ at the end of which $\hat{\mathsf{H}}$ outputs a bit b'. Let* transcript *denote all communication exchanged between $\hat{\mathsf{H}}$ and $\mathcal{A}$. If $g(\mathsf{transcript}) = w$ then H outputs the bit b' and else it outputs 0.*

Compared with selectivized game defined in Def. 14, H invokes a partially selective game $\hat{\mathsf{H}}$ to against the adversary $\mathcal{A}$ instead of invoking the adaptive game G. Note that although H requires $\mathcal{A}$ to commit $w \in \mathcal{W}$ at the beginning of the game, it only sends a partial message $u = h(w)$ to $\mathcal{A}$. The main idea of Jafargholi et al.'s framework [15] is to show that a sequence of selectivized games $\mathsf{SEL}_{\mathcal{W}}[\mathsf{G}, g]$ can be replaced by a sequence of further selectivized game $\mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}, g, h]$, and then our task is to guess $u \in \mathcal{U}$ rather than the whole $w \in \mathcal{W}$. In this way, we are able to avoid exponential security loss to a certain extent. Formally, we recall the following theorem from [15].

Theorem 8 ([15, Theorem 2]). *Let G_L and G_R be two adaptive games. For some function $g : \{0,1\}^* \to \mathcal{W}$ we define the selectivized games $\mathsf{H}_\mathsf{L} = \mathsf{SEL}_{\mathcal{W}}[\mathsf{G}_\mathsf{L}, g]$, $\mathsf{H}_\mathsf{R} = \mathsf{SEL}_{\mathcal{W}}[\mathsf{G}_\mathsf{R}, g]$. Let $\mathsf{H}_\mathsf{L} = \mathsf{H}_0, \mathsf{H}_1, \ldots, \mathsf{H}_\ell = \mathsf{H}_\mathsf{R}$ be some sequence of hybrid games. Assume that for each $i \in [0, \ell-1]$, there exists a function $h_i : \mathcal{W} \to \mathcal{U}$ and games $\hat{\mathsf{H}}_{i,0}, \hat{\mathsf{H}}_{i,1}$ such that:*

$$\mathsf{H}_i \equiv \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}_{i,0}, g, h_i] \quad , \quad \mathsf{H}_{i+1} \equiv \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}_{i,1}, g, h_i].$$

Furthermore, if for any PPT adversary $\mathcal{B}$ and any $i \in [0, \ell-1]$, it holds that

$$|\mathsf{Adv}_{\mathcal{B}}^{\hat{\mathsf{H}}_{i,0}} - \mathsf{Adv}_{\mathcal{B}}^{\hat{\mathsf{H}}_{i,1}}| \le \varepsilon,$$

then for any PPT adversary $\mathcal{A}$, we have that

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{G}_\mathsf{L}} - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{G}_\mathsf{R}}| \le \varepsilon \cdot \ell \cdot |\mathcal{U}|.$$

In summary, assuming that G_L and G_R are two adaptive game that we wish to prove indistinguishable, Jafargholi et al.'s framework [15] works as follows:

(1) Design $\mathsf{H}_\mathsf{L} = \mathsf{SEL}_{\mathcal{W}}[\mathsf{G}_\mathsf{L}, g]$ and $\mathsf{H}_\mathsf{R} = \mathsf{SEL}_{\mathcal{W}}[\mathsf{G}_\mathsf{R}, g]$ and prove selective security by a sequence of hybrid $\mathsf{H}_\mathsf{L} = \mathsf{H}_0, \mathsf{H}_1, \ldots, \mathsf{H}_\ell = \mathsf{H}_\mathsf{R}$.

(2) For each $i \in [0, \ell-1]$, design partially selective games $\hat{\mathsf{H}}_{i,0}, \hat{\mathsf{H}}_{i,1}$ such that:

$$\mathsf{H}_i \equiv \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}_{i,0}, g, h_i] \quad , \quad \mathsf{H}_{i+1} \equiv \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{H}}_{i,1}, g, h_i].$$

(3) Minimize the partial information, i.e., the size of $\mathcal{U}$, when we are proving the indistinguishability between $\hat{\mathsf{H}}_{i,0}$ and $\hat{\mathsf{H}}_{i,1}$ for each $i \in [0, \ell-1]$.

B.2 Proof of Theorem 1 (IND+wKP $\Rightarrow$ CPA)

In this proof, we will apply the framework proposed by Jafargholi et al. [15]. More precisely, we first design a sequence of fully-selective hybrids G_0-$\mathsf{G}_{\tau+1}$ and show the *selective* CPA security of an FPRE scheme based on its IND security and wKP security. Next, for each $\mathsf{G}_t, t \in [0,\tau]$, we design partially selective games $\hat{\mathsf{G}}_{t,0}, \hat{\mathsf{G}}_{t,1}$. Finally, we apply Theorem 8 to them and prove Theorem 1.

The fully-selective hybrids. The first step of our proof is to design the selectivized version of CPA experiment defined in Fig. 2. In fact, if we view users as vertices and re-encryption keys $\mathsf{rk}_{i\to j}^{f}$ that $\mathcal{A}$ obtains through $\mathcal{O}_{\text{ReKey}}$ queries as an edge from i to j, then users $[\mathfrak{n}]$ and $\mathcal{Q}_{rk}$ form a directed graph. For the directed graph, we define the subgraph that is reachable from the challenge user i^* as *the challenge graph*, denoted by G. In the selectivized game, we require the adversary to commit the challenge graph G that contains the challenger user i^* as well as all (honest) users reachable from i^* and the corresponding paths (re-encryption keys). More precisely, the selective experiment is thus defined as $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$, where $\mathcal{G}$ denotes the set of challenge graphs and g is the function that extracts the challenge graph $G \in \mathcal{G}$ from transcripts.

Our next step is to show that mFPRE is selectively secure, i.e., the advantage of any PPT adversary $\mathcal{A}$ against $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$ is negligible. Assuming that the adversary does not issue any re-encryption key queries from the challenge user i^* to other users, the FPRE scheme is the same as a PKE scheme with different levels of encryption algorithms. Thus, the CPA security follows directly from the IND security (cf. Def. 4) of mFPRE. More precisely, we can embed the challenge public key on the challenger user of CPA security

Unfortunately, any re-encryption key queries from the challenge user i^* to another user will cause this method to fail. This is because when we are trying to reduce CPA security to IND security, we do not know the secret key of i^*. As a result, if the adversary issues a query $\mathcal{O}_{\text{ReKey}}(i^*, j, f)$, we cannot invoke $\mathsf{FReKGen}(pk^{(i^*)}, sk^{(i^*)}, pk^{(j)}, f)$ to answer the query. Note that wKP security (cf. Def. 5) provides us with the ability to simulate re-encryption keys from user i to j without knowing $sk^{(i)}$. Therefore, before reducing CPA security to IND security, we need to arrive at a hybrid G_τ with $\tau \in \mathbb{N}$ where we are able to simulate re-encryption keys from i^* for the adversary without knowing $sk^{(i^*)}$.

To arrive at G_τ, we should make sure we can simulate $\mathsf{G}_{\tau-1}$ for $\mathcal{A}$ without knowledge of secret keys of $\mathsf{children}(i^*, G)$, where we embed the challenge user 0 in Fig. 5 to i^* and embed other users to $\mathsf{children}(i^*, G)$. Finally, wKP security implies that the advantage for $\mathcal{A}$ to distinguish $\mathsf{G}_{\tau-1}$ and G_τ is negligible.

The sequence of hybrids from G_0 (i.e., the original game $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$) to G_τ is exactly a pebbling sequence $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ for the challenge graph G. A pebbling configuration $\mathcal{P}_t \subseteq [\mathfrak{n}], t \in [0,\tau]$ is a set of pebbled vertices. A vertex i is pebbled in $\mathcal{P}_t$, i.e., $i \in \mathcal{P}_t$, means that in the hybrid G_t, we can answer the re-encryption queries with i as the source without the knowledge of its secret key $sk^{(i)}$. This pebbling game starts from a graph without any pebbles and ends with a graph where the challenge vertex i^* is pebbled, just like Def. 12. Note that if $\mathcal{P}$ is a valid sequence, the rules (cf. Def. 12) of pebbling game ensure that neighbouring hybrids are indistinguishable assuming wKP security.

Formally, we first show that IND security and wKP security imply selective CPA security via the following lemma.

Lemma 7. *For any PPT $\mathcal{A}$ against the selectivized game $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$ with challenge graph G (i.e., subgraph reachable from the vertex of challenge user) in $\mathcal{G}(\mathfrak{n},\delta,d)$ for $\mathfrak{n},\delta,d \in \mathbb{N}$, there exist PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ s.t. the advantage of $\mathcal{A}$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}},g]}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda) + 2\tau \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}(\lambda)$, where τ denote the time complexity for the class $\mathcal{G}(\mathfrak{n},\delta,d)$.*

Proof. Let $\mathcal{P} = (\mathcal{P}_0,\ldots,\mathcal{P}_\tau)$ be a pebbling sequence for the challenge graph G of $\mathcal{A}$. We prove the lemma via a sequence of game G_0-G_τ and $\mathsf{G}_{\tau+1}$, where G_t ($t \in [0,\tau]$) corresponds to the pebble configuration $\mathcal{P}_t$. More precisely, G_0 is the original selectivized game $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$ of $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ and $\mathcal{P}_0$ contains no pebbles, i.e., all re-encryption key queries are honestly generated. In $\mathcal{P}_\tau$, the source vertex/challenge user i^* is pebbled, which means that all re-encryption keys from i^* are fake (i.e., generated by simulation). Finally, in $\mathsf{G}_{\tau+1}$, we will show that the advantage of $\mathcal{A}$ is negligible assuming IND security.

<u>Game G_0:</u> This is the original selectivized game $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$, corresponding to the pebble configuration $\mathcal{P}_0$. Let Win denote the event that $\beta' = \beta$ in the case of $g(\mathsf{transcript}) = G$. By definition, $\mathsf{Adv}^{\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}},g]}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

At the beginning of this game, the adversary $\mathcal{A}$ commits a challenge graph $G \in \mathcal{G}(\mathfrak{n},\delta,d)$ to the challenger. According to G, the challenger computes a valid sequence of pebbling configurations $\mathcal{P} = (\mathcal{P}_0,\ldots,\mathcal{P}_\tau)$ with each $\mathcal{P}_t \subseteq [\mathfrak{n}]$.[6] G_0 is corresponding to $\mathcal{P}_0 = \emptyset$, i.e., all re-encryption keys are honestly generated via $\mathsf{FReKGen}$ by the challenger. Then the challenger answers $\mathcal{A}$'s queries just like $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ (cf. Fig. 2). At the end of the game, on receiving a bit β' from $\mathcal{A}$, the challenge extracts the real challenge graph $G' := g(\mathsf{transcript})$, where $\mathsf{transcript}$ denotes the transcripts throughout the game. If the real challenge graph G' is equal to the committed graph G (i.e., $G' = G$) and $\beta' = \beta$, the challenger outputs 1. Otherwise, the challenger outputs 0.

<u>Game $\mathsf{G}_t, t \in [\tau]$:</u> Game G_t is corresponding to the pebbling configuration $\mathcal{P}_t$. Each G_t is identical to G_0 except for the reply to $\mathcal{A}$'s re-encryption key queries $\mathcal{O}_{\textsc{ReKey}}(i,j,f)$.

– If $i \in \mathcal{P}_t$ and $j \in \mathsf{children}(i,G)$, the challenger invokes the simulation algorithm $\mathsf{rk}^f_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}^*(pk^{(i)},\, pk^{(j)}, f)$ guaranteed by the wKP security to generate $\mathsf{rk}^f_{i\to j}$, rather than invoking $\mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$.

– Otherwise, the challenger invokes $\mathsf{rk}^f_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$, just like G_0.

[6] We refer to [12, Algorithm 1] for an algorithm for computing pebbling sequence.

By the pebbling rules in Def. 12, we have that for each $t \in [\tau]$, $\exists!\ k \in \mathcal{P}_{t-1} \Delta \mathcal{P}_t$ and $\mathsf{children}(k, G) \subseteq \mathcal{P}_{t-1}$. Thus, we can reduce the indistinguishability between G_{t-1} and G_t to wKP security on user k. Formally, we have the following claim.

Claim 3. For all $t \in [\tau]$, $|\Pr_{t-1}[\mathsf{Win}] - \Pr_t[\mathsf{Win}]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}$.

Proof. We construct a PPT algorithm $\mathcal{B}'$ against the wKP security by simulating $\mathsf{G}_{t-1}/\mathsf{G}_t$ for $\mathcal{A}$ as follows.

Algorithm $\mathcal{B}'$. Algorithm $\mathcal{B}'$ is given the public keys $\{pk^{(i)}_{\mathsf{wKP}}\}_{i \in [0,\delta]}$ from its own challenger and has access to its own re-encryption key oracle $\mathcal{O}_{\text{ReKey}}(\cdot, \cdot)$. Let β_{wKP} denote the challenge bit chosen by $\mathcal{B}'$'s own challenger. $\mathcal{B}'$ wants to distinguish whether the re-encryption keys generated by its own oracle are real (i.e., $\beta_{\mathsf{wKP}} = 0$) or simulated (i.e., $\beta_{\mathsf{wKP}} = 1$).

$\mathcal{B}'$ is constructed by simulating $\mathsf{G}_{t-1}/\mathsf{G}_t$ for $\mathcal{A}$ as follows. At the beginning of the game, $\mathcal{B}'$ receives a challenge graph G from $\mathcal{A}$ and computes $\mathcal{P}_{t-1}, \mathcal{P}_t$. W.l.o.g., we assume that $|\mathcal{P}_{t-1}| \leq |\mathcal{P}_t|$. $\mathcal{B}'$ finds the unique user $k \in \mathcal{P}_{t-1} \Delta \mathcal{P}_t$ and its children $\mathsf{children}(k, G)$, then it embeds $pk^{(0)}_{\mathsf{wKP}}$ to $pk^{(k)}$ and $\{pk^{(i)}_{\mathsf{wKP}}\}_{i \in [\delta]}$ to $\{pk^{(j)}\}_{j \in \mathsf{children}(k,G)}$ (if $|\mathsf{children}(k, G)| < \delta$, then $\mathcal{B}'$ embeds the first $|\mathsf{children}(k, G)|$ public keys). For all other users $i \in [\mathfrak{n}] \setminus (\mathsf{children}(k, G) \cup \{k\})$, $\mathcal{B}'$ invokes KGen honestly to generate $(pk^{(i)}, sk^{(i)})$. $\mathcal{B}'$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, i^* = \perp$ and sends $\{pk^{(i)}\}_{i \in [\mathfrak{n}]}$ to $\mathcal{A}$.

- On receiving a re-encryption key query (i, j, f) from $\mathcal{A}$, if $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk} \cup \{(i, j)\}, \mathcal{Q}_c) = 1$, $\mathcal{B}'$ returns $\perp$ to $\mathcal{A}$, just like $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. Otherwise, $\mathcal{B}'$ replies the query as follows:
  - If $i \in \mathcal{P}_{t-1}$ and $j \in \mathsf{children}(i, G)$, $\mathcal{B}'$ invokes $\mathsf{rk}^f_{i \to j} \leftarrow_{\$} \mathsf{FReKGen}^*(pk^{(i)}, pk^{(j)}, f)$ and sends the simulated re-encryption key $\mathsf{rk}^f_{i \to j}$ to $\mathcal{A}$.
  - If $i = k$ and $j \in \mathsf{children}(k, G)$, $\mathcal{B}'$ queries (j, f) to its own oracle $\mathcal{O}_{\text{ReKey}}$. On receiving $\mathsf{rk}^f_{k \to j}$ from $\mathcal{O}_{\text{ReKey}}(j, f)$, $\mathcal{B}'$ passes it to $\mathcal{A}$.
  - Otherwise, $\mathcal{B}$ invokes $\mathsf{rk}^f_{i \to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$ and sends the real re-encryption key $\mathsf{rk}^f_{i \to j}$ to $\mathcal{A}$.

    Note that in the case of $i = k$ and $j \notin \mathsf{children}(k, G)$, $\mathcal{B}'$ cannot generate $\mathsf{rk}^f_{k \to j}$ without the knowledge of $sk^{(k)}$. But this case will lead to $g(\mathsf{transcript}) \neq G$ in both G_{t-1} and G_t.
- On receiving a corruption query i from $\mathcal{A}$, if $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$, $\mathcal{B}'$ returns $\perp$ to $\mathcal{A}$, just like $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. Otherwise, $\mathcal{B}'$ returns $sk^{(i)}$ to $\mathcal{A}$.

  Note that in the case of $i = k$ or $i \in \mathsf{children}(k, G)$, $\mathcal{B}$ does not possess $sk^{(i)}$. But this case will lead to $g(\mathsf{transcript}) \neq G$ in both G_{t-1} and G_t.
- On receiving the challenge tuple (i^*, m_0, m_1, v) from $\mathcal{A}$, if $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$, $\mathcal{B}'$ aborts the game with $\mathcal{A}$ and returns a random bit $\beta'_{\mathsf{wKP}} \leftarrow_{\$} \{0, 1\}$ to its own challenger. Otherwise, $\mathcal{B}'$ chooses a random bit β for $\mathcal{A}$ and returns $ct^*_v \leftarrow_{\$} \mathsf{Enc}(pk^{(i^*)}, m_\beta, v)$ to $\mathcal{A}$.
- Finally, $\mathcal{B}'$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}'$ outputs $\beta'_{\mathsf{wKP}} = 1$ to its own challenger if and only if $\beta' = \beta$ and $g(\mathsf{transcript}) = G$. Otherwise, $\mathcal{B}'$ outputs a uniform bit $\beta'_{\mathsf{wKP}} \leftarrow_{\$} \{0, 1\}$.

Note that in the case of $g(\mathsf{transcript}) \neq G$, $\mathcal{B}'$ will outputs a uniform bit β'_{wKP} in both G_{t-1} and G_t and this will have no effect on the difference in probability of the event Win.

Now we analyze the advantage of $\mathcal{B}'$ in the case of $g(\mathsf{transcript}) = G$. If the challenge bit $\beta_{\mathsf{wKP}} = 0$, i.e., $\mathcal{B}'$'s own $\mathcal{O}_{\text{ReKey}}(\cdot,\cdot)$ oracle always returns real re-encryption keys, $\mathcal{B}'$ simulates G_{t-1} perfectly for $\mathcal{A}$. If the challenge bit $\beta_{\mathsf{wKP}} = 1$, i.e., $\mathcal{B}'$'s own $\mathcal{O}_{\text{ReKey}}(\cdot,\cdot)$ oracle always returns simulated re-encryption keys, $\mathcal{B}'$ simulates G_t perfectly for $\mathcal{A}$. Thus,

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}(\lambda) = |\Pr[\beta'_{\mathsf{wKP}} = \beta_{\mathsf{wKP}}] - \tfrac{1}{2}| \\
&= \tfrac{1}{2} \cdot |\Pr[\beta'_{\mathsf{wKP}} = 1 \mid \beta_{\mathsf{wKP}} = 0] - \Pr[\beta'_{\mathsf{wKP}} = 1 \mid \beta_{\mathsf{wKP}} = 1]| \\
&= \tfrac{1}{2} \cdot |\Pr[\beta' = \beta \wedge g(\mathsf{transcript}) = G \mid \beta_{\mathsf{wKP}} = 0] \;- \\
&\qquad\qquad\qquad \Pr[\beta' = \beta \wedge g(\mathsf{transcript}) = G \mid \beta_{\mathsf{wKP}} = 1]| \\
&= \tfrac{1}{2} \cdot |\Pr_{t-1}[\mathsf{Win}] - \Pr_t[\mathsf{Win}]|.
\end{aligned}$$

This completes the proof of Claim 3. ▌

Game G_τ: Game G_τ is corresponding to the pebbling configuration $\mathcal{P}_\tau$. By Claim 3 and a simple hybrid, we have

$$|\Pr_0[\mathsf{Win}] - \Pr_\tau[\mathsf{Win}]| \leq 2\tau \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}.$$

Note that in G_τ, the challenge user i^* is pebbled, which means the secret key $sk^{(i^*)}$ is not needed anymore.

Game $\mathsf{G}_{\tau+1}$: It is the same as G_τ, except for the generation of the challenge ciphertext ct^*_v. Now the challenger always encrypts m_1, i.e., $ct^*_v \leftarrow_\$ \mathsf{Enc}(pk^{(i^*)}, m_1, v)$, regardless of the challenge bit β.

We show the computational indistinguishability between G_τ and $\mathsf{G}_{\tau+1}$ via the following claim.

Claim 4. $|\Pr_\tau[\mathsf{Win}] - \Pr_{\tau+1}[\mathsf{Win}]| \leq \mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{B}}$.

Proof. We construct a PPT algorithm $\mathcal{B}$ against the IND security by simulating $\mathsf{G}_\tau/\mathsf{G}_{\tau+1}$ for $\mathcal{A}$ as follows.

Algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ is given a public key pk from its own challenger and has access to its own challenge oracle $\mathcal{O}_{\text{Chal}}(\cdot,\cdot,\cdot)$. $\mathcal{B}$ wants to guess the challenge bit β_{IND} chosen by its own challenger.

$\mathcal{B}$ is constructed by simulating $\mathsf{G}_\tau/\mathsf{G}_{\tau+1}$ for $\mathcal{A}$ as follows. At the beginning of the game, $\mathcal{B}$ receives a challenge graph G from $\mathcal{A}$ and computes $\mathcal{P}_\tau$. $\mathcal{B}$ embeds pk to the public key $pk^{(i^*)}$ of the challenge user i^*. For all other users $i \in [\mathfrak{n}] \setminus \{i^*\}$, $\mathcal{B}$ invokes KGen honestly to generate $(pk^{(i)}, sk^{(i)})$. $\mathcal{B}$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, i^* = \perp$ and sends $\{pk^{(i)}\}_{i\in[\mathfrak{n}]}$ to $\mathcal{A}$.

- On receiving a re-encryption key query (i, j, f) from $\mathcal{A}$, if $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, just like $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. Otherwise, $\mathcal{B}$ replies the query as follows:

- If $i \in \mathcal{P}_\tau$ and $j \in \mathsf{children}(i, G)$, $\mathcal{B}$ invokes $\mathsf{rk}_{i\to j}^{f} \leftarrow_{\$} \mathsf{FReKGen}^*(pk^{(i)}, pk^{(j)}, f)$ and sends the simulated re-encryption key $\mathsf{rk}_{i\to j}^{f}$ to $\mathcal{A}$.
- Otherwise, $\mathcal{B}$ invokes $\mathsf{rk}_{i\to j}^{f} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$ and sends the real re-encryption key $\mathsf{rk}_{i\to j}^{f}$ to $\mathcal{A}$.

  Note that in the case of $i = i^*$ and $j \notin \mathsf{children}(i, G)$, $\mathcal{B}$ cannot generate $\mathsf{rk}_{i^*\to j}^{f}$ without the knowledge of $sk^{(i^*)}$. But this case will lead to $g(\mathsf{transcript}) \neq G$ in both G_τ and $\mathsf{G}_{\tau+1}$.

– On receiving a corruption query i from $\mathcal{A}$, if $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, just like $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathsf{n}}^{\mathsf{CPA}}$. Otherwise, $\mathcal{B}$ returns $sk^{(i)}$ to $\mathcal{A}$.

  Note that in the case of $i = i^*$, $\mathcal{B}$ does not possess $sk^{(i^*)}$. But this case will lead to $g(\mathsf{transcript}) \neq G$ in both G_τ and $\mathsf{G}_{\tau+1}$.

– On receiving the challenge tuple (i^*, m_0, m_1, v) from $\mathcal{A}$, if $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$, $\mathcal{B}$ aborts the game with $\mathcal{A}$ and returns a random bit $\beta'_{\mathsf{IND}} \leftarrow_{\$} \{0, 1\}$ to its own challenger. Otherwise, $\mathcal{B}$ queries (m_0, m_1, v) to its own challenge oracle $\mathcal{O}_{\textsc{Chal}}$ and receives ct_v. Then $\mathcal{B}$ chooses a random bit $\beta \leftarrow_{\$} \{0, 1\}$ for $\mathcal{A}$. In the case of $\beta = 0$, $\mathcal{B}$ sets $ct_v^* := ct_v$ and in the case of $\beta = 1$, $\mathcal{B}$ generates $ct_v^* \leftarrow_{\$} \mathsf{Enc}(pk^{(i^*)}, m_1, v)$ itself. $\mathcal{B}$ returns ct_v^* to $\mathcal{A}$.

– Finally, $\mathcal{B}$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}$ outputs $\beta'_{\mathsf{IND}} = 1$ to its own challenger if and only if $\beta' = \beta$ and $g(\mathsf{transcript}) = G$. Otherwise, $\mathcal{B}$ outputs a uniform bit $\beta'_{\mathsf{IND}} \leftarrow_{\$} \{0, 1\}$.

Note that in the case of $g(\mathsf{transcript}) \neq G$, $\mathcal{B}$ will outputs a uniform bit β'_{IND} in both G_τ and $\mathsf{G}_{\tau+1}$ and this will have no effect on the difference in probability of the event Win.

Now we analyze the advantage of $\mathcal{B}$ in the case of $g(\mathsf{transcript}) = G$. If the challenge bit $\beta_{\mathsf{IND}} = 0$, i.e., $\mathcal{B}$'s own challenge oracle $\mathcal{O}_{\textsc{Chal}}(m_0, m_1, v)$ always encrypts m_0, $\mathcal{B}$ simulates G_τ perfectly for $\mathcal{A}$. If the challenge bit $\beta_{\mathsf{IND}} = 1$, i.e., $\mathcal{B}$'s own challenge oracle $\mathcal{O}_{\textsc{Chal}}(m_0, m_1, v)$ always encrypts m_1, $\mathcal{B}$ simulates $\mathsf{G}_{\tau+1}$ perfectly for $\mathcal{A}$. Thus,

$$\begin{aligned}
&\mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}}^{\mathsf{IND}}(\lambda) = |\Pr[\beta'_{\mathsf{IND}} = \beta_{\mathsf{IND}}] - \tfrac{1}{2}| \\
&= \tfrac{1}{2} \cdot |\Pr[\beta'_{\mathsf{IND}} = 1 \mid \beta_{\mathsf{IND}} = 0] - \Pr[\beta'_{\mathsf{IND}} = 1 \mid \beta_{\mathsf{IND}} = 1]| \\
&= \tfrac{1}{2} \cdot |\Pr[\beta' = \beta \wedge g(\mathsf{transcript}) = G \mid \beta_{\mathsf{IND}} = 0] \;- \\
&\qquad\qquad \Pr[\beta' = \beta \wedge g(\mathsf{transcript}) = G \mid \beta_{\mathsf{IND}} = 1]| \\
&= \tfrac{1}{2} \cdot |\Pr_\tau[\mathsf{Win}] - \Pr_{\tau+1}[\mathsf{Win}]|.
\end{aligned}$$

This completes the proof of Claim 4. ▌

Finally, note that in $\mathsf{G}_{\tau+1}$, the challenge bit β is completely hidden to $\mathcal{A}$, thus we have $\Pr_{\tau+1}[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Lemma 7 follows. □

The partially-selective hybrids. For any two neighboring hybrids G_t and G_{t+1}, $t \in [0, \tau - 1]$, the only difference between them is the unique vertex $k = \mathcal{P}_t \Delta \mathcal{P}_{t+1}$. Consequently, to simulate $\mathsf{G}_t/\mathsf{G}_{t+1}$ for the adversary $\mathcal{A}$, we do not need the whole challenge graph G, but the set of pebbled vertices $\mathcal{P}_t$, one vertex k that needs to be pebbled (or unpebbled) together with its children $\mathsf{children}(k, G)$[7]. Note that once the adversary queries a re-encryption key query from user k to user j (regardless of whether j belongs to $\mathsf{children}(k, G)$), we always return the simulated re-encryption key. Then there are two cases:

- There exists a re-encryption key query $\mathcal{O}_{\textsc{ReKey}}(k, j, f)$ s.t. $j \notin \mathsf{children}(k, G)$. In this case, the real challenge graph $g(\mathsf{transcript})$ differs from the committed graph G, and the challenger always outputs a random bit b.
- Otherwise, the simulation of $\mathsf{G}_t/\mathsf{G}_{t+1}$ is perfect.

Thus, we can define $\mathcal{U}$ as the set of elements $(\mathcal{P}_t, k, \mathsf{children}(k, G))$ and have

$$|\mathcal{U}| \leq |\mathcal{V}|^{\sigma+\delta+1} = \mathfrak{n}^{\sigma+\delta+1},$$

where $\sigma := \max_{t\in[0,\tau]} |\mathcal{P}_t|$ denotes the space complexity of the pebbling sequence $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ and δ the outdegree of G.

Now, for any $\mathsf{G}_t, \mathsf{G}_{t+1}$ $(t \in [0, \tau])$ used in the proof of Lemma 7, we present the further selectivized games $\mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{G}}_{t,0}, g, h_t], \mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{G}}_{t,1}, g, h_t]$ and the partially selective hybrids $\hat{\mathsf{G}}_{t,0}, \hat{\mathsf{G}}_{t,1}$ as below.

Further Selectivized Game $\mathsf{SEL}_{\mathcal{U}\to\mathcal{W}}[\hat{\mathsf{G}}_{t,\hat{b}}, g, h_t], t \in [0, \tau], \hat{b} \in \{0, 1\}$: At the beginning of the game, the adversary $\mathcal{A}$ commits a challenge graph $G \in \mathcal{G}(\mathfrak{n}, \delta, d)$ to the challenger. According to G, the challenger first computes a valid sequence of pebbling configurations $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_\tau)$ with each $\mathcal{P}_t \subseteq [\mathfrak{n}]$ and computes the partial information $h_t(G) := (\mathcal{P}_t, k, \mathsf{children}(k, G))$ as follows:

- In the case of $t \leq \tau - 1$, it finds the unique $k := \mathcal{P}_t \varDelta \mathcal{P}_{t+1}$ along with its children $\mathsf{children}(k, G)$.
- In the case of $t = \tau$, it sets k to the challenge user i^*, and $\mathsf{children}(k, G) = \emptyset$.

Then the challenger runs the partially-selective game $\hat{\mathsf{G}}_{t,\hat{b}}(\mathcal{P}_t, k, \mathsf{children}(k, G))$ as defined below:

Partially-selective Games $\hat{\mathsf{G}}_{t,\hat{b}}(\mathcal{P}_t, k, \mathsf{children}(k, G)), t \in [0, \tau], \hat{b} \in \{0, 1\}$: If $t \leq \tau - 1$, this game is the same as the (adaptive) CPA experiment $\mathsf{Exp}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$, except for the reply to $\mathcal{A}$'s re-encryption key queries $\mathcal{O}_{\textsc{ReKey}}(i, j, f)$ in the case that no trivial attacks occur. Here, w.l.o.g., we assume that $k \notin \mathcal{P}_t$, i.e., k is vertex that needs to be pebbled.[8]

- If $i \in \mathcal{P}_t$, the challenger invokes $\mathsf{FReKGen}^*(pk^{(i)}, pk^{(j)}, f)$ to generate $\mathsf{rk}^f_{i\to j}$ and returns the simulated re-encryption key to the adversary.
- If $i = k$ and $\hat{b} = 1$, the challenger invokes $\mathsf{FReKGen}^*(pk^{(k)}, pk^{(j)}, f)$ to generate $\mathsf{rk}^f_{k\to j}$ and returns the simulated re-encryption key to the adversary.

[7] Note that we need the information $\mathsf{children}(k, G)$ to embed δ public keys when reducing the difference of $\hat{\mathsf{G}}_{t,0}/\hat{\mathsf{G}}_{t,1}$ to the wKP security (cf. the proof of Claim 3).

[8] If $k \in \mathcal{P}_t$, then we need to unpebble k, i.e., convert the reply of re-encryption key queries from simulated re-encryption keys to real re-encryption keys.

- If $i = k$ and $\hat{b} = 0$, the challenger invokes $\mathsf{FReKGen}(pk^{(k)}, sk^{(k)}, pk^{(j)}, f)$ to generate $\mathsf{rk}_{k \to j}^{f}$ and returns the real re-encryption key to the adversary.
- Otherwise, the challenger invokes $\mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$ to generate $\mathsf{rk}_{i \to j}^{f}$ and returns the real re-encryption key to the adversary.

On receiving the output bit b' from $\hat{\mathsf{G}}_{t,\hat{b}}(\mathcal{P}_t, k)$, the challenger checks if $g(\mathsf{transcript}) = G$. In the case of $g(\mathsf{transcript}) = G$, the challenger outputs $b := b'$, otherwise, the challenger outputs a uniform bit $b \leftarrow_{\$} \{0,1\}$. It is easy to verify that for any PPT adversary $\mathcal{A}$, we have $|\mathsf{Adv}_{\mathcal{A}}^{\hat{\mathsf{G}}_{t,0}}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\hat{\mathsf{G}}_{t,1}}(\lambda)| \leq 2 \cdot \mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}',\delta}^{\mathsf{wKP}}(\lambda)$. The proof is similar to the proof of Claim 3.

If $t = \tau$, $\hat{\mathsf{G}}_{\tau,0}$ is identical to $\hat{\mathsf{G}}_{\tau-1,1}$, and $\hat{\mathsf{G}}_{\tau,1}$ is almost identical to $\hat{\mathsf{G}}_{\tau,0}$ except that the challenge ciphertext is always generated by $ct_v^* \leftarrow_{\$} \mathsf{Enc}(pk^{(i^*)}, m_1, v)$, regardless of β. It is easy to verify that for any PPT adversary $\mathcal{A}$, we have $|\mathsf{Adv}_{\mathcal{A}}^{\hat{\mathsf{G}}_{\tau,0}}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\hat{\mathsf{G}}_{\tau,1}}(\lambda)| \leq 2 \cdot \mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}}^{\mathsf{IND}}(\lambda)$. The proof is similar to the proof of Claim 4.

Finally, for each $t \in [0, \tau]$, we have $\mathsf{G}_t \equiv \mathsf{SEL}_{\mathcal{G} \to \mathcal{V}^{\sigma+1}}[\hat{\mathsf{G}}_{t,0}, g, h_t]$ and $\mathsf{G}_{t+1} \equiv \mathsf{SEL}_{\mathcal{G} \to \mathcal{V}^{\sigma+1}}[\hat{\mathsf{G}}_{t,1}, g, h_t]$ hold for the $h_t : \mathcal{G} \to \mathcal{V}^{\sigma+\delta+1}$ and partially-selective games $\hat{\mathsf{G}}_{t,0}, \hat{\mathsf{G}}_{t,1}$ defined above. By applying Theorem 8, we complete the proof of Theorem 1. □

B.3 Proof of Theorem 2 (IND+wKP+SH ⇒ HRA)

The proof of Theorem 2 consists of two main steps. In the first step, we define an intermediate security notion, i.e., shHRA security, via the experiment $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}$ illustrated in Fig. 11. Roughly speaking, the challenger will answer $\mathcal{A}$'s re-encryption queries $\mathcal{O}_{\text{ReEnc}}(i, j, f, k)$ with freshly generated ciphertexts, instead of re-encrypted ciphertexts, in the case $(k, i) \notin \mathcal{L}^*$, i.e., the ciphertext to be re-encrypted is not (derivative of) the challenge ciphertext ct_v^*. At the end of the first step, we will show the indistinguishability of $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}}$ and $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}$. Note that in $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}$, the $\mathcal{O}_{\text{Enc}}$ oracle and $\mathcal{O}_{\text{ReEnc}}$ do not leak any information of the challenge bit β beyond the challenge ciphertext ct_v^* to $\mathcal{A}$ and the answers no longer involve $sk^{(i^*)}$ to compute the re-encryption keys for those $(k, i) \notin \mathcal{L}^*$. Then, in the second step, we prove the shHRA security of mFPRE in a similar way as the strategy we proved the CPA security in Appendix B.2.

We first present the formal definition of the intermediate security notion, i.e., the shHRA security of mFPRE.

Definition 16 **(shHRA Security for Multi-Hop FPRE).** *A multi-hop FPRE scheme* mFPRE *is* shHRA *secure, if for any PPT adversary $\mathcal{A}$ and any polynomial $\mathfrak{n}$, it holds that* $\mathsf{Adv}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}(\lambda) := \left|\Pr[\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$, *where the experiment* $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}$ *is defined in Fig. 11.*

Compared with the HRA experiment $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{HRA}}$ (cf. Fig. 3), the first difference in $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{shHRA}}$ is that the challenger uses an extra column in $\mathcal{L}$ to store

```
$\underline{\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}}$:
For $i \in [\mathfrak{n}]$: $(pk^{(i)}, sk^{(i)}) \leftarrow_\$ \mathsf{KGen}$
$\mathcal{Q}_{rk} := \emptyset$                    //record re-encryption key queries
$\mathcal{Q}_c := \emptyset$                         //record corruption queries
$i^* := \perp$                                       //record challenge user
$\mathcal{L} := \perp$                    //record honestly generated ciphertexts
$\mathcal{L}^* := \perp$        //record derivatives of the challenge ciphertext
$\mathsf{ctr} := 0$                //index of honestly generated ciphertexts
$(i^*, m_0, m_1, v, st) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{Cor}}(\cdot),\mathcal{O}_{\text{Enc}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{ReEnc}}(\cdot,\cdot,\cdot,\cdot)}$
                    $(\{pk^{(i)}\}_{i\in[\mathfrak{n}]})$
If $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$:
    Return $b \leftarrow_\$ \{0,1\}$                          //avoid TA1, TA2
$\beta \leftarrow_\$ \{0,1\}$
$\mathsf{ctr} := \mathsf{ctr} + 1$
$ct_v^* \leftarrow_\$ \mathsf{Enc}(pk^{(i^*)}, m_\beta, v)$
$\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, i^*, m_\beta, (ct_v^*, v))\}$
$\mathcal{L}^* := \mathcal{L}^* \cup \{(\mathsf{ctr}, i^*)\}$            //index of challenge ciphertext
$\beta' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{ReKey}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{Cor}}(\cdot),\mathcal{O}_{\text{Enc}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{ReEnc}}(\cdot,\cdot,\cdot,\cdot)}(st, ct_v^*)$

If $\beta' = \beta$: Return 1;   Else: Return 0
```

```
$\underline{\mathcal{O}_{\text{ReKey}}(i,j,f)}$:                    //re-encryption key queries
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$:
        Return $\perp$                                            //avoid TA2
    $\mathcal{Q}_{rk} := \mathcal{Q}_{rk} \cup \{(i,j)\}$
    $\mathsf{rk}^f_{i\to j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
    Return $\mathsf{rk}^f_{i\to j}$
```

```
$\underline{\mathcal{O}_{\text{Cor}}(i)}$:                                 //corruption queries
    If $\exists(i,\cdot) \in \mathcal{L}^*$:  Return $\perp$         //avoid TA1, TA3
    If $\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$:
        Return $\perp$                                         //avoid TA2
    $\mathcal{Q}_c := \mathcal{Q}_c \cup \{i\}$
    Return $sk^{(i)}$
```

```
$\underline{\mathcal{O}_{\text{Enc}}(i,m,v)}$:                //honest encryption queries
    $\mathsf{ctr} := \mathsf{ctr} + 1$
    $ct_v^{(i)} \leftarrow_\$ \mathsf{Enc}(pk^{(i)}, m, v)$
    $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, i, m, (ct_v^{(i)}, v))\}$
    Return $(\mathsf{ctr}, ct_v^{(i)})$
```

```
$\underline{\mathcal{O}_{\text{ReEnc}}(i,j,f,k)}$:        //honest re-encryption queries
    If $(k,i) \in \mathcal{L}^*$ and $j \in \mathcal{Q}_c$: Return $\perp$//avoid TA3
    Retrieve $(k, i, m, (ct', v'))$ from $\mathcal{L}$:
        If fails, return $\perp$
    If $(k,i) \notin \mathcal{L}^*$ :
        $ct^{(j)}_{v'+1} \leftarrow_\$ \mathsf{Enc}(pk^{(j)}, f(m), v'+1)$
    Else:
        $\mathsf{rk}^f_{i\to j} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$
        $ct^{(j)}_{v'+1} \leftarrow_\$ \mathsf{FReEnc}(\mathsf{rk}^f_{i\to j}, ct', v')$
    $\mathsf{ctr} := \mathsf{ctr} + 1$
    $\mathcal{L} := \mathcal{L} \cup \{(\mathsf{ctr}, j, f(m), (ct^{(j)}_{v'+1}, v'+1))\}$
    If $(k,i) \in \mathcal{L}^*$:   $\mathcal{L}^* := \mathcal{L}^* \cup \{(\mathsf{ctr}, j)\}$
    Return $(\mathsf{ctr}, ct^{(j)}_{v'+1})$
```

```
$\underline{\mathsf{CheckTA}(i^*, \mathcal{Q}_{rk}, \mathcal{Q}_c)}$:                      //check TA2
    If $\exists\ (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$
        s.t. $j_t \in \mathcal{Q}_c$ for some $t \geq 1$:
        Return 1
    Else: Return 0
```

Fig. 11. The shHRA security experiment $\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for mFPRE. For ease of reading, we emphasize different parts with the HRA experiment $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ in gray boxes.

the underlying message m, which is intended for the change of the re-encryption oracle $\mathcal{O}_{\text{ReEnc}}$. The second difference in $\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ is that the challenger now answers $\mathcal{A}$'s re-encryption queries $\mathcal{O}_{\text{ReEnc}}(i,j,f,k)$ with freshly generated ciphertexts $ct^{(j)}_{v'+1}$ in the case $(k,i) \notin \mathcal{L}^*$, i.e., the ciphertext to be re-encrypted is not (derivative of) the challenge ciphertext ct_v^*.

Next we prove the indistinguishability between $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ and $\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ based on the SH security via the following lemma.

Lemma 8. *For any PPT adversary $\mathcal{A}$ and $\mathfrak{n} \in \mathbb{N}$, there exists a PPT $\mathcal{B}$ s.t.*

$$|\mathsf{Adv}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) - \mathsf{Adv}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda)| \leq 2\mathfrak{n}(\mathfrak{n}-1)L \cdot \mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}}.$$

Proof. We begin by introducing a new notion named *ciphertext freshness* (from $\mathcal{A}$'s view), denoted by $\mathsf{Fresh}(ct) \in [0, L]$ for ciphertext ct. The freshness of a ciphertext $ct_v^{(i)}$ generated by the $\mathcal{O}_{\text{Enc}}(i,m,v)$ oracle is defined as $\mathsf{Fresh}(ct_v^{(i)}) := 0$, regardless of its level v. If a ciphertext $ct^{(j)}_{v'+1}$ is generated by $\mathcal{O}_{\text{ReEnc}}(i,j,f,k)$ and the ciphertext ct' indexed by k (i.e., the ct' contained in the retrieval

$(k, i, m, (ct', v'))$ from $\mathcal{L}$) has freshness s, then $\mathsf{Fresh}(ct^{(j)}_{v'+1}) := s + 1$. For any ciphertext generated by $\mathcal{O}_{\text{Enc}}$ or $\mathcal{O}_{\text{ReEnc}}$, the challenger can easily obtain its freshness by recording the freshness of them at the moment of answering the queries. In addition, we say a ciphertext is *fresh* if it is generated by the Enc algorithm.

Now consider the ciphertexts that $\mathcal{A}$ obtains from the challenger and let $\mathcal{C}^{(i)}_s$ denote the set of ciphertexts encrypted under i's public key with freshness s, where $i \in [\mathfrak{n}]$ and $s \in [0, L]$. Note that in $\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$, except for the re-encryptions of the challenge ciphertext ct^*_v, all ciphertexts obtained by $\mathcal{A}$ is fresh, although the freshness of them may be different.

As an example, in the case $L = 2$, we show possible relations between $\mathcal{C}^{(i)}_s$ with $i \in [\mathfrak{n}]$ and $s \in [0, 2]$ in Fig. 12. Taking $\mathcal{C}^{(2)}_1$ as an example, ciphertexts $ct \in \mathcal{C}^{(2)}_1$ may be re-encryptions of ciphertexts in $\mathcal{C}^{(1)}_0$ or $\mathcal{C}^{(j)}_0$, and some of $ct \in \mathcal{C}^{(2)}_1$ may be further re-encrypted to ciphertexts in $\mathcal{C}^{(1)}_2$.

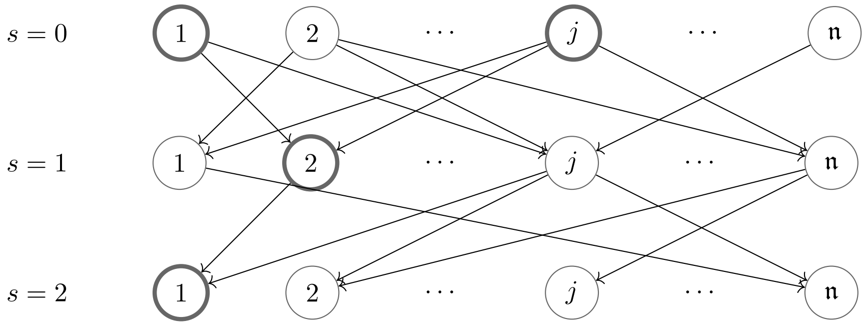

Fig. 12. Possible relations between all ciphertexts the $\mathcal{A}$ obtains from the challenger in the case of $L = 2$. The round node $i \in [\mathfrak{n}]$ in the row of $s \in [0, 2]$ denotes the set $\mathcal{C}^{(i)}_s$ of ciphertexts encrypted under $pk^{(i)}$ with freshness s. An arrow from node i to j means that $\mathcal{A}$ has queried $\mathcal{O}_{\text{ReEnc}}(i, j, \cdot, \cdot)$.

Our final goal is to replace ciphertexts in $\mathcal{C}^{(i)}_s$ with fresh ciphertexts for all $i \in [\mathfrak{n}]$ and $s \in [0, L]$. The graph shown in Fig. 12 can help us replace the re-encryptions accurately, however, this graph is clear only at the very end of the experiment $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}/\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ since $\mathcal{A}$ can adaptively issue its re-encryption queries. As a result, we have to replace *all possible* queries by fresh ciphertexts, instead of real re-encryptions.

SH security (cf. Def. 6) can help us accomplish the replacement above, since it guarantees that $\mathcal{A}$ cannot tell whether a ciphertext $ct^{(j)}_{v'+1}$ is a re-encryption of another ciphertext $ct^{(i)}_{v'}$ or is freshly encrypted. Note that the SH security requires the underlying ciphertext $ct^{(i)}_{v'}$ to be freshly encrypted. Thus we will perform the replacement row by row. More precisely, we first replace all ciphertexts with freshness $s = 1$ by fresh ciphertexts, then replace all ciphertexts with freshness $s = 2$ and so on, ending with highest level $s = L$.

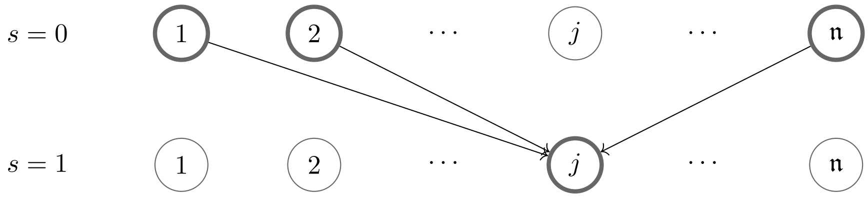

Fig. 13. Users $i \in [\mathfrak{n}] \setminus \{j\}$ we need to consider when replacing ciphertexts $\mathcal{C}_1^{(j)}$ with fresh generated ciphertexts.

Before we replace all ciphertexts with freshness 1, let us first consider to replace ciphertexts with freshness $s = 1$ under user j's public key, i.e., $\mathcal{C}_1^{(j)}$. Ciphertexts in $\mathcal{C}_1^{(j)}$ have $\mathfrak{n} - 1$ possible sources as shown in Fig. 13 and we have no idea about $\mathcal{A}$'s choices. Thus, for all re-encryption queries targeting user j, we replace the real re-encryptions by fresh ciphertexts source by source.

For each user couple (i, j), the SH security makes sure that the change is indistinguishable to $\mathcal{A}$. Informally, when reducing to the SH security, the reduction algorithm $\mathcal{B}$ will embed user 0 and user 1 in the SH experiment to user i and user j, respectively. When $\mathcal{A}$ issues $\mathcal{O}_{\textsc{Enc}}$ and $\mathcal{O}_{\textsc{ReEnc}}$ queries, $\mathcal{B}$ will send them to its own $\mathcal{O}_{\textsc{Enc}}$ and $\mathcal{O}_{\textsc{Chal}}$ oracles and simply pass the answer to $\mathcal{A}$. For other queries, as $sk^{(0)}$ and $sk^{(1)}$ are known to the reduction algorithm $\mathcal{B}$, $\mathcal{B}$ can generate the answers itself.

Below we present a formal description of a sequence of games G_0-$\mathsf{G}_{\mathfrak{n}.\mathfrak{n}-1.L}$ with $\mathsf{G}_0 = \mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ and $\mathsf{G}_{\mathfrak{n}.\mathfrak{n}-1.L} = \mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$, and describe the reduction algorithm $\mathcal{B}$.

<u>**Game** G_0:</u> This is the HRA experiment $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ (cf. Fig. 3). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Next we present $\mathfrak{n}(\mathfrak{n} - 1)L$ games and each game is labeled by $\mathfrak{j}, \mathfrak{i}, s$ with $\mathfrak{j} \in [\mathfrak{n}], \mathfrak{i} \in [\mathfrak{n}] \setminus \{\mathfrak{j}\}, s \in [L]$. The first game is $\mathsf{G}_{1,2,1}$ and the order of games is defined as:

(1) Increment $\mathfrak{i}$ until running out of the set $[\mathfrak{n}] \setminus \{\mathfrak{j}\}$.
(2) Increment $\mathfrak{j}$ until $\mathfrak{j} = \mathfrak{n}$ and reset value of $\mathfrak{i} \in [\mathfrak{n}] \setminus \{\mathfrak{j}\}$. Now all ciphertexts in $\mathcal{C}_s^{(\mathfrak{j})}$ have been replaced by fresh ciphertexts and we move to the next user $\mathfrak{j} + 1$.
(3) Increment s until $s = L$ and reset the values of $\mathfrak{j} \in [\mathfrak{n}]$ and $\mathfrak{i} \in [\mathfrak{n}] \setminus \{\mathfrak{j}\}$. Now all ciphertexts with freshness s have been replaced by fresh ciphertexts and we move to the next freshness $s + 1$.

<u>**Game** $\mathsf{G}_{\mathfrak{j}.\mathfrak{i}.s}, \mathfrak{j} \in [\mathfrak{n}], \mathfrak{i} \in [\mathfrak{n}] \setminus \{\mathfrak{j}\}, s \in [L]$:</u> It is the same as the previous game (denoted by G_{pre}), except for the reply to $\mathcal{A}$'s re-encryption queries $\mathcal{O}_{\textsc{ReEnc}}(i, j, f, k)$.

- If $i = \mathfrak{i}$, $j = \mathfrak{j}$ and the challenger successfully retrieves $(k, \mathfrak{i}, m, (ct', v'))$ from $\mathcal{L}$ with $\mathsf{Fresh}(ct') = s-1$, the challenger returns $ct^{(\mathfrak{j})}_{v'+1} \leftarrow_\$ \mathsf{Enc}(pk^{(\mathfrak{j})}, f(m), v'+1)$ instead of the real re-encrypted ciphertext.
- Otherwise, the challenger answers in the same way as in G_{pre}.

Claim 5. For all $\mathfrak{j} \in [\mathfrak{n}], \mathfrak{i} \in [\mathfrak{n}] \setminus \{\mathfrak{j}\}, s \in [L]$, $|\Pr_{pre}[\mathsf{Win}] - \Pr_{\mathfrak{j}.\mathfrak{i}.s}[\mathsf{Win}]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda)$.

Proof. We construct a PPT algorithm $\mathcal{B}$ against the SH security by simulating $\mathsf{G}_{pre}/\mathsf{G}_{\mathfrak{j}.\mathfrak{i}.s}$ for $\mathcal{A}$ as follows.

Algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ is given $(pk^{(0)}_{\mathsf{SH}}, sk^{(0)}_{\mathsf{SH}}, pk^{(1)}_{\mathsf{SH}}, sk^{(1)}_{\mathsf{SH}})$ from its own challenger and has access to its own oracles $\mathcal{O}_{\text{ReKey}}, \mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Chal}}$. $\mathcal{B}$ wants to guess the challenge bit β_{SH} chosen by its own challenger.

$\mathcal{B}$ is constructed by simulating $\mathsf{G}_{pre}/\mathsf{G}_{\mathfrak{j}.\mathfrak{i}.s}$ for $\mathcal{A}$ as follows. For user $\mathfrak{i}$ and $\mathfrak{j}$, $\mathcal{B}$ sets $pk^{(\mathfrak{i})} := pk^{(0)}_{\mathsf{SH}}, sk^{(\mathfrak{i})} := sk^{(0)}_{\mathsf{SH}}$ and $pk^{(\mathfrak{j})} := pk^{(1)}_{\mathsf{SH}}, sk^{(\mathfrak{j})} := sk^{(1)}_{\mathsf{SH}}$. For all other users $i \in [\mathfrak{n}] \setminus \{\mathfrak{i}, \mathfrak{j}\}$, $\mathcal{B}$ invokes KGen honestly to generate $(pk^{(i)}, sk^{(i)})$. Note that $\mathcal{B}$ owns the secret keys $\{sk^{(i)}\}_{i\in[\mathfrak{n}]}$ of all users. Then $\mathcal{B}$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, i^* = \perp, \mathcal{L} = \emptyset, \mathcal{L}^* = \emptyset, \mathsf{ctr} = 0$ and sends $\{pk^{(i)}\}_{i\in[\mathfrak{n}]}$ to $\mathcal{A}$.

- On receiving the a re-encryption key query (i, j, f) from $\mathcal{A}$ that leads to no trivial attack, if $i = \mathfrak{i}$ and $j = \mathfrak{j}$, $\mathcal{B}$ queries $\mathcal{O}_{\text{ReKey}}(f)$ to its own challenger and returns the received $\mathsf{rk}^f_{\mathfrak{i}\to\mathfrak{j}}$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ answers the query in the same way as in $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.
- On receiving a honest encryption query (i, m, v) from $\mathcal{A}$, if $i = \mathfrak{i}$, $\mathcal{B}$ queries $\mathcal{O}_{\text{Enc}}(m, v)$ to its own challenger and receives $(\mathsf{ctr}_{\mathsf{SH}}, ct)$. $\mathcal{B}$ labels ct with its own counter ctr, stores $(\mathsf{ctr}, \mathfrak{i}, m, (ct, v))$ to $\mathcal{L}$ and returns $(\mathsf{ctr}, ct^{(\mathfrak{i})}_v := ct)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ answers the query in the same way as in $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.
- On receiving a re-encryption query (i, j, f, k) from $\mathcal{A}$ that leads to no trivial attack, $\mathcal{B}$ first retrieves $(k, i, m, (ct', v'))$ from $\mathcal{L}$. If $i = \mathfrak{i}, j = \mathfrak{j}, (k, \mathfrak{i}) \notin \mathcal{L}^*$ and $\mathsf{Fresh}(ct') = s-1$, it finds out the corresponding index $\mathsf{ctr}_{\mathsf{SH}}$ and queries $\mathcal{O}_{\text{Chal}}(\mathsf{ctr}_{\mathsf{SH}}, f)$ to its own challenger. On receiving ct from $\mathcal{B}$'s challenger, $\mathcal{B}$ labels ct with its own counter ctr, stores $(\mathsf{ctr}, \mathfrak{j}, m, ct, v'+1))$ to $\mathcal{L}$ and returns $(\mathsf{ctr}, ct^{(\mathfrak{j})}_{v'+1} := ct)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ answers the query in the same way as in $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.
- On receiving a corruption query i from $\mathcal{A}$, $\mathcal{B}$ answers the query just as defined in $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.
- On receiving the challenge tuple (i^*, m_0, m_1, v), $\mathcal{B}$ answers $\mathcal{A}$ in the same way as in $\mathsf{Exp}^{\mathsf{HRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.
- Finally, $\mathcal{B}$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}$ outputs $\beta'_{\mathsf{SH}} = 1$ to its own challenger if and only if $\beta' = \beta$.

Now we analyze the advantage of $\mathcal{B}$. Overall, if $\beta_{\mathsf{SH}} = 0$, $\mathcal{B}$ simulates G_{pre} perfectly for $\mathcal{A}$, and if $\beta_{\mathsf{SH}} = 1$, $\mathcal{B}$ simulates $\mathsf{G}_{\mathfrak{j},\mathfrak{i},s}$ perfectly for $\mathcal{A}$. Thus,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda) &= \tfrac{1}{2} \cdot |\Pr[\beta'_{\mathsf{SH}} = 1 \mid \beta_{\mathsf{SH}} = 0] - \Pr[\beta'_{\mathsf{SH}} = 1 \mid \beta_{\mathsf{SH}} = 1]| \\ &= \tfrac{1}{2} \cdot |\Pr_{pre}[\mathsf{Win}] - \Pr_{\mathfrak{j}.\mathfrak{i}.s}[\mathsf{Win}]|. \end{aligned}$$ ∎

Note that in $\mathsf{G}_{\mathfrak{n}.\mathfrak{n}-1.L}$, all ciphertexts that $\mathcal{A}$ obtains from the challenger have been replaced by fresh ciphertexts. Thus, we have $\mathsf{Adv}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) = |\Pr_{\mathfrak{n},\mathfrak{n}-1,L}[\mathsf{Win}] - \frac{1}{2}|$.

Finally, by Claim 5 and a simple hybrid argument, we complete the proof of Lemma 8. □

In the second part of the proof, we show that mFPRE is selectively shHRA secure if it has both IND security and wKP security via the following lemma.

Lemma 9. *For any PPT $\mathcal{A}$ against the selectivized game $\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}, g]$ with challenge graph G (i.e., subgraph reachable from the vertex of challenge user) in $\mathcal{G}(\mathfrak{n},\delta,d)$, there exist PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ s.t. the advantage of $\mathcal{A}$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SEL}_{\mathcal{G}}[\mathsf{Exp}^{\mathsf{shHRA}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}},g]}(\lambda) \leq 2 \cdot \mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda) + 2\tau \cdot \mathsf{Adv}^{\mathsf{wKP}}_{\mathsf{mFPRE},\mathcal{B}',\delta}(\lambda)$, where τ denote the time complexity for the class $\mathcal{G}(\mathfrak{n},\delta,d)$.*

Proof sketch. Now if the adversary issues a re-encryption query $\mathcal{O}_{\textsc{ReEnc}}(i,j,f,k)$ and $(k,i) \in \mathcal{L}^*$, an edge (i,j) will be added to the challenge graph. The remaining proof of Lemma 9 is identical to the proof of Lemma 7 since the extra oracles $\mathcal{O}_{\textsc{Enc}}$ and $\mathcal{O}_{\textsc{ReEnc}}$ do not leak any other information[9] about the challenge bit β beyond the challenge ciphertext ct_v^*.

The construction of the partially selective hybrids and analysis are similar to that of the CPA security in Appendix B.2. □

Finally, combining Lemma 8, Lemma 9 and Theorem 8 together, we complete the proof of Theorem 2. □

C Correctness and Fine-Grained L-Hop Correctness of mFPRE_2

Correctness. For a v-level ciphertext ct_v generated by $\mathsf{Enc}(pk,\mathbf{m},v)$, we have $ct_v = \binom{\overline{ct_v}}{\underline{ct_v}} = \binom{\overline{\mathbf{A}}\mathbf{s}+\overline{\mathbf{e}}}{\underline{\mathbf{A}}\mathbf{s}+\underline{\mathbf{e}}+p\mathbf{m}}$, where $\mathbf{e} = \binom{\overline{\mathbf{e}}}{\underline{\mathbf{e}}} \leftarrow_{\$} \chi_v^{N+\ell}$. Since $\mathbf{e} \leftarrow_{\$} \chi_v^{N+\ell}$ is B_v-bounded and we have $B_v \leq B_L < \min\{p/2, q/(10N)\}$ for all $v \in [0,L]$, we can show that the decryption algorithm Dec recovers $\mathbf{m}$ correctly from ct_v by a similar analysis as that for mFPRE_1 in Subsect. 4.1.

Fine-Grained L-Hop Correctness. For $ct_0^{(i)} \xrightarrow{\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}}} ct_1^{(j)}$, where $ct_0^{(i)} \leftarrow_{\$} \mathsf{Enc}(pk^{(i)},\mathbf{m},0)$, $\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f_{\mathbf{M}_1})$ and $ct_1^{(j)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}_{i\to j}^{f_{\mathbf{M}_1}}, ct_0^{(i)}, 0)$, we will show that the decryption of $ct_1^{(j)}$ results in $f_{\mathbf{M}_1}(\mathbf{m}) = \mathbf{M}_1\mathbf{m}$. Note that $ct_1^{(j)} := \hat{ct}_1^{(j)} + \mathbf{A}^{(j)}\mathbf{s}_1' + \mathbf{e}_1'$ with $\hat{ct}_1^{(j)} := \mathsf{rk}_{i\to j}^{f_{\mathbf{M}}} \cdot ct_0^{(i)}$ and

[9] Note that in the last game, the challenger re-encrypts challenge ciphertexts with the simulated re-encryption key.

$\mathbf{s}'_1 \leftarrow_{\$} \chi_1^n, \mathbf{e}'_1 \leftarrow_{\$} \chi_1^{N+\ell}$. With a similar analysis as that for mFPRE_1 in Subsect. 4.1, i.e., (10), we have that

$$\hat{ct}_1^{(j)} = \mathbf{A}^{(j)} \underbrace{\mathbf{S}\mathbf{s}_0}_{:=\mathbf{s}_1} + \underbrace{\mathbf{E}\mathbf{s}_0 + \mathbf{R}_1\overline{\mathbf{e}_0} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\underline{\mathbf{e}_0}\end{pmatrix}}_{:=\mathbf{e}_1} + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_1}(\mathbf{m})}\end{pmatrix},$$

where $\mathbf{s}_0 \leftarrow_{\$} \chi_0^n$, $\mathbf{e}_0 = \binom{\overline{\mathbf{e}_0}}{\underline{\mathbf{e}_0}} \leftarrow_{\$} \chi_0^{N+\ell}$, $\mathbf{S} \leftarrow_{\$} \chi^{n\times n}$, $\mathbf{E} \leftarrow_{\$} \chi^{(N+\ell)\times n}$, and it follows that $\|\mathbf{s}_1\|_\infty \le nBB_0$ and $\|\mathbf{e}_1\|_\infty \le (nB + NB + \ell\Delta)B_0$. Consequently,

$$ct_1^{(j)} = \hat{ct}_1^{(j)} + \mathbf{A}^{(j)}\mathbf{s}'_1 + \mathbf{e}'_1 = \mathbf{A}^{(j)} \cdot (\underbrace{\mathbf{s}_1 + \mathbf{s}'_1}_{:=\mathbf{s}''_1}) + \underbrace{\mathbf{e}_1 + \mathbf{e}'_1}_{:=\mathbf{e}''_1} + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_1}(\mathbf{m})}\end{pmatrix},$$

and we have $\|\mathbf{s}''_1\|_\infty \le \|\mathbf{s}_1\|_\infty + \|\mathbf{s}'_1\|_\infty \le nBB_0 + B_1 \le 2B_1$ and $\|\mathbf{e}''_1\|_\infty \le \|\mathbf{e}_1\|_\infty + \|\mathbf{e}'_1\|_\infty \le (nB+NB+\ell\Delta)B_0 + B_1 \le 2B_1 \le 2B_L < \min\{p/2, q/(10N)\}$. Therefore, the decryption algorithm Dec recovers $f_{\mathbf{M}_1}(\mathbf{m}) = \mathbf{M}_1\mathbf{m}$ from $ct_1^{(j)}$.

As the re-encryption proceeds, after L hops of re-encryption under $f_{\mathbf{M}_1}, f_{\mathbf{M}_2}, \cdots, f_{\mathbf{M}_L}$, we can get an L-level ciphertext $ct_L^{(\eta)}$ and it satisfies

$$\hat{ct}_L^{(\eta)} = \mathbf{A}^{(\eta)} \underbrace{\mathbf{S}\mathbf{s}''_{L-1}}_{:=\mathbf{s}_L} + \underbrace{\mathbf{E}\mathbf{s}''_{L-1} + \mathbf{R}_L\overline{\mathbf{e}''_{L-1}} + \begin{pmatrix}\mathbf{0}\\ \mathbf{M}_1\underline{\mathbf{e}''_{L-1}}\end{pmatrix}}_{:=\mathbf{e}_L} + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_L\cdots\mathbf{M}_2\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_L}(\cdots f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m})))}\end{pmatrix},$$

$$ct_L^{(\eta)} = \hat{ct}_L^{(\eta)} + \mathbf{A}^{(\eta)}\mathbf{s}'_L + \mathbf{e}'_L = \mathbf{A}^{(\eta)} \cdot (\underbrace{\mathbf{s}_L + \mathbf{s}'_L}_{:=\mathbf{s}''_L}) + \underbrace{\mathbf{e}_L + \mathbf{e}'_L}_{:=\mathbf{e}''_L} + \begin{pmatrix}\mathbf{0}\\ p\cdot \underbrace{\mathbf{M}_L\cdots\mathbf{M}_2\mathbf{M}_1\mathbf{m}}_{=f_{\mathbf{M}_L}(\cdots f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m})))}\end{pmatrix},$$

where $\|\mathbf{s}_L\|_\infty \le nB \cdot 2B_{L-1}$, $\|\mathbf{e}_L\|_\infty \le (nB + NB + \ell\Delta) \cdot 2B_{L-1}$ and $\mathbf{s}'_L \leftarrow_{\$} \chi_L^n$, $\mathbf{e}'_L \leftarrow_{\$} \chi_L^{N+\ell}$. Then we have $\|\mathbf{s}''_L\|_\infty \le nB \cdot 2B_{L-1} + B_L \le 2B_L$ and $\|\mathbf{e}''_L\|_\infty \le (nB + NB + \ell\Delta) \cdot 2B_{L-1} + B_L \le 2B_L < \min\{p/2, q/(10N)\}$, and consequently, the function value $f_{\mathbf{M}_L}(\cdots f_{\mathbf{M}_2}(f_{\mathbf{M}_1}(\mathbf{m}))) = \mathbf{M}_L \cdots \mathbf{M}_2\mathbf{M}_1\mathbf{m}$ can be recovered from $ct_L^{(\eta)}$ by the decryption algorithm Dec.

D Omitted Proofs

D.1 Proof of Claim 1

Proof of Claim 1. Firstly, we construct a PPT adversary $\mathcal{B}'$ against the Q_{chal}-$\mathsf{LWE}_{n,q,\chi,N+\ell}$-assumption, such that $\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \le \mathsf{Adv}^{Q_{chal}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda)$. Then by a standard hybrid argument, we have $\mathsf{Adv}^{Q_{chal}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda) \le Q_{chal} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}}(\lambda)$ and the claim follows.

Algorithm $\mathcal{B}'$. Given a challenge $(\mathbf{A}, \mathbf{U})$, $\mathcal{B}'$ wants to distinguish $\mathbf{U} = \mathbf{AS} + \mathbf{E}$ from $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$, where $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times n}$, $\mathbf{S} \leftarrow_{\$} \chi^{n\times Q_{chal}}$, $\mathbf{E} \leftarrow_{\$} \chi^{(N+\ell)\times Q_{chal}}$.

$\mathcal{B}'$ is constructed by simulating $\mathsf{G}_0/\mathsf{G}_1$ for $\mathcal{A}$ as follows. Firstly, $\mathcal{B}'$ sets the public key $pk := \mathbf{A}$ and returns pk to $\mathcal{A}$. Then $\mathcal{B}'$ chooses a random bit

$\beta \leftarrow_{\$} \{0,1\}$ and parses $\mathbf{U} = (\mathbf{u}_1 \mid \cdots \mid \mathbf{u}_{Q_{chal}}) \in \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$ with each $\mathbf{u}_k \in \mathbb{Z}_q^{N+\ell}$ for $k \in [Q_{chal}]$. On $\mathcal{A}$'s k-th $\mathcal{O}_{\text{CHAL}}(\mathbf{m}_0, \mathbf{m}_1, v)$ query ($k \in [Q_{chal}]$), $\mathcal{B}'$ computes $ct_v := \mathbf{u}_k + \binom{\mathbf{0}}{p\mathbf{m}_\beta}$ and returns ct_v to $\mathcal{A}$. Finally, $\mathcal{B}'$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}'$ outputs 1 to its own challenger if and only if $\beta' = \beta$.

Now we analyze the advantage of $\mathcal{B}'$.

- In the case of $\mathbf{U} = \mathbf{AS} + \mathbf{E}$, by parsing $\mathbf{S} = (\mathbf{s}_1 \mid \cdots \mid \mathbf{s}_{Q_{chal}}) \in \mathbb{Z}_q^{n\times Q_{chal}}$ with each $\mathbf{s}_k \in \mathbb{Z}_q^n$ and parsing $\mathbf{E} = (\mathbf{e}_1 \mid \cdots \mid \mathbf{e}_{Q_{chal}}) \in \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$ with each $\mathbf{e}_k \in \mathbb{Z}_q^{N+\ell}$, we have $\mathbf{u}_k = \mathbf{A}\mathbf{s}_k + \mathbf{e}_k$ for $\mathbf{s}_k \leftarrow_{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_k \leftarrow_{\$} \mathbb{Z}_q^{N+\ell}$ for all $k \in [Q_{chal}]$. Consequently, the ciphertext simulated by $\mathcal{B}'$ is $ct_v := \mathbf{u}_k + \binom{\mathbf{0}}{p\mathbf{m}_\beta} = \mathbf{A}\mathbf{s}_k + \mathbf{e}_k + \binom{\mathbf{0}}{p\mathbf{m}_\beta}$, the same as that in G_0.
- In the case of $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$, each $\mathbf{u}_k$ is uniformly distributed over $\mathbb{Z}_q^{N+\ell}$ for $k \in [Q_{chal}]$. Therefore, the ciphertext $ct_v := \mathbf{u}_k + \binom{\mathbf{0}}{p\mathbf{m}_\beta}$ simulated by $\mathcal{B}'$ is also uniformly distributed over $\mathbb{Z}_q^{N+\ell}$, the same as that in G_1.

Overall, $\mathcal{B}'$ simulates G_0 for $\mathcal{A}$ in the case $\mathbf{U} = \mathbf{AS} + \mathbf{E}$ and simulates G_1 for $\mathcal{A}$ in the case $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$. Thus $\mathcal{B}'$ successfully distinguishes $\mathbf{U} = \mathbf{AS} + \mathbf{E}$ from $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{(N+\ell)\times Q_{chal}}$ as long as the probability that $\beta' = \beta$ in G_0 differs non-negligibly from that in G_1. Consequently, we have $\mathsf{Adv}^{Q_{chal}\text{-}\mathsf{LWE}}_{[n,q,\chi,N+\ell],\mathcal{B}'}(\lambda) \geq \big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big|$, as desired. This completes the proof of Claim 1. ▌

D.2 Proof of Theorem 5

Proof of Theorem 5. We prove the theorem via a sequence of games $\mathsf{G}_0 - \mathsf{G}_{L+1}$.

Game G_0: This is the IND experiment (cf. Fig. 4). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{IND}}_{\mathsf{mFPRE}_2,\mathcal{A}}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Let $(pk = \mathbf{A}, sk = \mathbf{T})$. In this game, the challenger chooses a random bit $\beta \leftarrow_{\$} \{0,1\}$ and answers $\mathcal{A}$'s $\mathcal{O}_{\text{CHAL}}$ queries $(\mathbf{m}_0, \mathbf{m}_1, v)$ with $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, \mathbf{m}_\beta, v)$, i.e., $ct_v := \mathbf{As} + \mathbf{e} + \binom{\mathbf{0}}{p\mathbf{m}_\beta}$ for $\mathbf{s} \leftarrow_{\$} \chi_v^n, \mathbf{e} \leftarrow_{\$} \chi_v^{N+\ell}$.

Game $\mathsf{G}_t, t \in [L+1]$: It is the same as G_0, except for the reply to $\mathcal{A}$'s $\mathcal{O}_{\text{CHAL}}(\mathbf{m}_0, \mathbf{m}_1, v)$ query:

- For $v \leq t-1$ the challenger returns a uniformly sampled $ct_v \leftarrow_{\$} \mathbb{Z}_q^{N+\ell}$ to $\mathcal{A}$.
- For $v \geq t$, the challenger answers the query just like G_0, that is, $ct_v \leftarrow_{\$} \mathsf{Enc}(pk, \mathbf{m}_\beta, v)$.

Note that the only difference between G_{t-1} and G_t is the reply to $\mathcal{A}$'s $\mathcal{O}_{\text{CHAL}}(\mathbf{m}_0, \mathbf{m}_1, v)$ queries when $v = t-1$: in G_{t-1}, the challenger answers the queries with real ciphertexts $ct_{t-1} := \mathbf{As}+\mathbf{e}+\binom{\mathbf{0}}{p\mathbf{m}_\beta}$ where $\mathbf{s} \leftarrow_{\$} \chi_{t-1}^n, \mathbf{e} \leftarrow_{\$} \chi_{t-1}^{N+\ell}$, while in G_t, the challenger answers with random ciphertexts $ct_{t-1} \leftarrow_{\$} \mathbb{Z}_q^{N+\ell}$. Thus, assuming that $\mathcal{A}$ makes at most Q_{chal} queries to $\mathcal{O}_{\text{CHAL}}$, the difference between G_{t-1} and G_t can be reduced to the Q_{chal}-$\mathsf{LWE}_{n,q,\chi_{t-1},N+\ell}$-assumption. Formally, we have the following claim. Its proof is essentially the same as that for Claim 1 in the proof of Theorem 3, thus we omit it.

Claim 6. For all $t \in [L+1]$, $|\Pr_{t-1}[\mathsf{Win}] - \Pr_t[\mathsf{Win}]| \le Q_{chal} \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi_{t-1},N+\ell],\mathcal{B}_{t-1}}(\lambda)$.

Finally, note that in G_{L+1}, all $\mathcal{O}_{\text{CHAL}}(\mathbf{m}_0, \mathbf{m}_1, v)$ queries are answered with random ciphertexts $ct_v \leftarrow_\$ \mathbb{Z}_q^{N+\ell}$, thus the challenge bit β is completely hidden to $\mathcal{A}$, and we have $\Pr_{L+1}[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Theorem 5 follows. □

E More Discussions on Other Security Notions for Multi-Hop FPRE in Subsect. 3.3

E.1 More Discussions on UNID Security (Def. 7) and Its Relation to CPA Security

Remark 4 (On the formalization of UNID *security and discussion on trivial attacks).* Unidirectionality of a multi-hop FPRE scheme requires that given a re-encryption key $\mathsf{rk}^{f}_{j^* \to i^*}$, it is hard for the adversary to come up with re-encryption key $\mathsf{rk}^{f'}_{i^* \to j^*}$ of the other direction even if the adversary is able to obtain some re-encryption keys and corrupt some users to obtain their secret keys.

We note that there might not exist a specialized PPT algorithm to check whether $\mathsf{rk}^{f'}_{i^* \to j^*}$ is indeed a re-encryption key from i^* to j^*. Thus in $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$, we actually check the *functionality* of $\mathsf{rk}^{f'}_{i^* \to j^*}$, i.e., whether it can convert a (0-level) ciphertext of user i^* that encrypts a randomly chosen message m into a (1-level) ciphertext of user j^* that encrypts $f'(m)$.

Actually, there are five trivial attacks **TA1′-TA5′** to obtain $\mathsf{rk}^{f'}_{i^* \to j^*}$ or obtain the functionality of $\mathsf{rk}^{f'}_{i^* \to j^*}$ for some f'.

TA1′: $i^* = j^*$, in this case, $\mathcal{A}$ directly gets $\mathsf{rk}^{f'}_{i^* \to j^*} = \mathsf{rk}^{f}_{j^* \to i^*}$ for $f' = f$.

TA2′: $i^* \in \mathcal{Q}_c$, i.e., $\mathcal{A}$ ever obtains $sk^{(i^*)}$. In this case, $\mathcal{A}$ can use $sk^{(i^*)}$ to generate $\mathsf{rk}^{f'}_{i^* \to j^*}$ itself by invoking $\mathsf{rk}^{f'}_{i^* \to j^*} \leftarrow_\$ \mathsf{FReKGen}(pk^{(i^*)}, sk^{(i^*)}, pk^{(j^*)}, f')$.

TA3′: $\exists (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$ s.t. $j_t \in \mathcal{Q}_c$ for some $t \ge 1$, i.e., $\mathcal{A}$ gets a chain of re-encryption keys $\mathsf{rk}^{f_1}_{i^* \to j_1}, \mathsf{rk}^{f_2}_{j_1 \to j_2}, \ldots, \mathsf{rk}^{f_t}_{j_{t-1} \to j_t}$ starting from user i^* and ending at some corrupted user j_t for whom $\mathcal{A}$ ever obtains its secret key $sk^{(j_t)}$. In this case, $\mathcal{A}$ can invoke $\mathsf{rk}^{f_{t+1}}_{j_t \to j^*} \leftarrow_\$ \mathsf{FReKGen}(pk^{(j_t)}, sk^{(j_t)}, pk^{(j^*)}, f_{t+1})$, so that the chain of re-encryption keys from i^* to j_t is further extended to j^*, i.e.,

$$\mathsf{rk}^{f_1}_{i^* \to j_1}, \mathsf{rk}^{f_2}_{j_1 \to j_2}, \ldots, \mathsf{rk}^{f_t}_{j_{t-1} \to j_t}, \mathsf{rk}^{f_{t+1}}_{j_t \to j^*}.$$

This new chain of re-encryption keys achieves the same functionality of $\mathsf{rk}^{f'}_{i^* \to j^*}$ with $f' = f_{t+1} \circ f_t \circ \cdots \circ f_2 \circ f_1$.

TA4$'$: $\exists (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$ s.t. $j_t = j^*$ for some $t \geq 1$, i.e., $\mathcal{A}$ gets a chain of re-encryption keys $\mathsf{rk}^{f_1}_{i^* \to j_1}, \mathsf{rk}^{f_2}_{j_1 \to j_2}, \ldots, \mathsf{rk}^{f_t}_{j_{t-1} \to j^*}$ starting from user i^* and ending at user j^*. In this case, this chain already achieves the same functionality of $\mathsf{rk}^{f'}_{i^* \to j^*}$ with $f' = f_t \circ \cdots \circ f_2 \circ f_1$.

TA5$'$: The function f' is a constant function or an almost constant function, i.e., f' maps (almost) all messages $m \in \mathcal{M}$ to a constant $c = f'(m) \in \mathcal{M}$. In this case, the functionality of $\mathsf{rk}^{f'}_{i^* \to j^*}$ can be approximated by $\mathsf{Enc}(pk^{(j^*)}, c) = \mathsf{Enc}(pk^{(j^*)}, f'(m))$.

To exclude **TA5$'$**, we require that the function f' for which $\mathcal{A}$ produces $\mathsf{rk}^{f'}_{i^* \to j^*}$ satisfies the property of *output diversity*, i.e.,

$$\Pr[m_0, m_1 \leftarrow_\$ \mathcal{M} : f'(m_0) \neq f'(m_1)] \geq 1/\mathsf{poly}(\lambda). \tag{14}$$

In [24], Zhou et al. formally proved that the CPA security implies the UNID security in the single-hop setting. Below we show that the relation $\mathsf{CPA} \Rightarrow \mathsf{UNID}$ also holds for multi-hop FPRE.

Lemma 10 (CPA $\Rightarrow$ UNID for Multi-Hop FPRE). *If a multi-hop FPRE scheme* mFPRE *is* CPA *secure, then it is also* UNID *secure.*

Proof of Lemma 10. To prove this, we show that if there exists a PPT adversary $\mathcal{A}$ breaking the unidirectionality (UNID) of mFPRE, then we can construct a PPT algorithm $\mathcal{B}$ to break the CPA security by simulating $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ for $\mathcal{A}$.

Algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ is given the public keys $\{pk^{(i)}\}_{i \in [\mathfrak{n}]}$ from its own challenger and has access to its own oracles $\mathcal{O}_{\text{ReKey}}, \mathcal{O}_{\text{Cor}}$.

(1) $\mathcal{B}$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, i^* = \perp, j^* = \perp$ and sends $\{pk^{(i)}\}_{i \in [\mathfrak{n}]}$ to $\mathcal{A}$.

- On receiving a re-encryption key query (i, j, f) from $\mathcal{A}$, $\mathcal{B}$ checks $\mathcal{A}$'s trivial attacks by checking if $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk} \cup \{(i, j)\}, \mathcal{Q}_c) = 1$, just like $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, otherwise $\mathcal{B}$ adds (i, j) to $\mathcal{Q}_{rk}$ and queries (i, j, f) to its own oracle $\mathcal{O}_{\text{ReKey}}$. On receiving $\mathsf{rk}^{f}_{i \to j}$ from $\mathcal{O}_{\text{ReKey}}(i, j, f)$, $\mathcal{B}$ returns $\mathsf{rk}^{f}_{i \to j}$ to $\mathcal{A}$.
- On receiving a corruption query i from $\mathcal{A}$, $\mathcal{B}$ checks $\mathcal{A}$'s trivial attacks by checking if $i = i^*$ or $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$, just like $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, otherwise $\mathcal{B}$ adds i to $\mathcal{Q}_c$ and queries i to its own oracle $\mathcal{O}_{\text{Cor}}$. On receiving $sk^{(i)}$ from $\mathcal{O}_{\text{Cor}}(i)$, $\mathcal{B}$ returns $sk^{(i)}$ to $\mathcal{A}$.
- On receiving the challenge tuple (i^*, j^*, f) from $\mathcal{A}$, $\mathcal{B}$ first checks if $(i^* = j^*)$ or $(i^* \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(i^*, j^*, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$ to identify trivial attacks, just like $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If yes, $\mathcal{B}$ aborts the experiment with $\mathcal{A}$ and returns a random bit $\beta' \leftarrow_\$ \{0, 1\}$ to its own challenger. Otherwise, $\mathcal{B}$ adds (j^*, i^*) to $\mathcal{Q}_{rk}$, and queries j^* to its own oracle $\mathcal{O}_{\text{Cor}}$. On receiving $sk^{(j^*)}$ from $\mathcal{O}_{\text{Cor}}(j^*)$, $\mathcal{B}$ invokes $\mathsf{rk}^{f}_{j^* \to i^*} \leftarrow_\$ \mathsf{FReKGen}(pk^{(j^*)}, sk^{(j^*)}, pk^{(i^*)}, f)$ and return $\mathsf{rk}^{f}_{j^* \to i^*}$ to $\mathcal{A}$.

(2) Finally, on receiving $\mathcal{A}$'s answer $(f', \mathsf{rk}^{f'}_{i^* \to j^*})$, $\mathcal{B}$ checks whether f' has output diversity efficiently. If f' does not have output diversity, $\mathcal{B}$ aborts the experiment with $\mathcal{A}$ and returns a random bit $\beta' \leftarrow_{\$} \{0,1\}$ to its own challenger. Otherwise, $\mathcal{B}$ chooses $m_0, m_1 \leftarrow_{\$} \mathcal{M}$ s.t. $f'(m_0) \neq f'(m_1)$, and sends challenge tuple $(i^*, m_0, m_1, 0)$ to its own challenger.

On receiving $ct_0^{(i^*)}$ from its own challenger, $\mathcal{B}$ invokes $ct_1^{(j^*)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f'}_{i^* \to j^*}, ct_0^{(i^*)}, 0)$ using the $\mathsf{rk}^{f'}_{i^* \to j^*}$ produced by $\mathcal{A}$ and computes $m' := \mathsf{Dec}(sk^{(j^*)}, ct_1^{(j^*)})$. If $m' = f'(m_0)$, $\mathcal{B}$ sets $\beta' = 0$, and if $m' = f'(m_1)$, $\mathcal{B}$ sets $\beta' = 1$, otherwise, $\mathcal{B}$ picks a random bit $\beta' \leftarrow_{\$} \{0,1\}$. $\mathcal{B}$ returns β' to its own challenger.

In the simulation, if $\mathcal{A}$ implements trivial attacks **TA1′-TA5′**, $\mathcal{B}$ will abort the experiment, just like $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. Otherwise, no trivial attacks from $\mathcal{A}$ implies that $i^* \notin \mathcal{Q}_c$ and there does not exist a chain of re-encryption keys from i^* to $j \in \mathcal{Q}_c \cup \{j^*\}$, while $\mathcal{Q}_c \cup \{j^*\}$ is exactly the corrupted users set for $\mathcal{B}$'s challenger. Thus, $\mathcal{B}$ never issue queries leading to trivial attacks **TA1** and **TA2**. So $\mathcal{B}$ is able to simulate $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ perfectly for $\mathcal{A}$.

Now we analyze the advantage of $\mathcal{B}$. Note that $\mathcal{A}$ wins in $\mathsf{Exp}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ means that the $\mathsf{rk}^{f'}_{i^* \to j^*}$ produced by $\mathcal{A}$ passes the check of functionality. Therefore, in the case of $\mathcal{A}$ wins, for the challenge ciphertext $ct_0^{(i^*)}$ that encrypts the randomly chosen message m_β, the re-encrypted ciphertext $ct_1^{(j^*)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f'}_{i^* \to j^*}, ct_0^{(i^*)}, 0)$ using the $\mathsf{rk}^{f'}_{i^* \to j^*}$ produced by $\mathcal{A}$ will decrypt to $m' := \mathsf{Dec}(sk^{(j^*)}, ct_1^{(j^*)}) = f'(m_\beta)$, and thus $\mathcal{B}$ can guess β correctly with probability 1. Otherwise, $\mathcal{B}$ will submit a random bit β' to its own challenger, and thus guess β correctly with probability 1/2. Overall,

$$\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B},\mathfrak{n}}(\lambda) = |\Pr[\beta' = \beta] - \tfrac{1}{2}|$$
$$= |\Pr[\mathcal{A} \text{ wins}] \cdot \Pr[\beta' = \beta \mid \mathcal{A} \text{ wins}] + \Pr[\neg\mathcal{A} \text{ wins}] \cdot \Pr[\beta' = \beta \mid \neg\mathcal{A} \text{ wins}] - \tfrac{1}{2}|$$
$$= |\Pr[\mathcal{A} \text{ wins}] \cdot 1 + (1 - \Pr[\mathcal{A} \text{ wins}]) \cdot \tfrac{1}{2} - \tfrac{1}{2}| = \tfrac{1}{2} \cdot \Pr[\mathcal{A} \text{ wins}] = \tfrac{1}{2} \cdot \mathsf{Adv}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda).$$

Consequently, if $\mathcal{A}$ breaks the unidirectionality (UNID) of mFPRE with a non-negligible advantage $\mathsf{Adv}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda)$, then $\mathcal{B}$ will break the CPA security with a non-negligible advantage $\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B},\mathfrak{n}}(\lambda) = \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{UNID}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda)$ as well. □

E.2 More Discussions on CUL Security (Def. 8)

Remark 5 (On the formalization of CUL *security and discussion on trivial attacks).* We formalize the CUL security by defining the experiment $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$ in Fig. 8. Similar to previous security notions, we consider a multi-user setting, and the adversary $\mathcal{A}$ is allowed to make $\mathcal{O}_{\text{ReKey}}$ and $\mathcal{O}_{\text{Cor}}$ queries *adaptively* to obtain re-encryption keys and secret keys, respectively. At some point, $\mathcal{A}$ outputs a set of challenge users $\mathcal{Q}_u = \{i_j\}_{j \in [0,L]}$, a sequence of functions $\{f_j\}_{j \in [L]}$ together with a message m, as well as a sequence of messages $\{m_j\}_{j \in [0,L]}$, and receives a set of challenge ciphertexts $(ct_0^{(i_0)}, ct_1^{(i_1)}, \ldots, ct_L^{(i_L)})$ which is

- (Case $\beta = 0$) *either* a chain of ciphertexts generated by re-encryption hops, i.e., $ct_0^{(i_0)} \leftarrow_\$ \mathsf{Enc}(pk^{(i_0)}, m, 0)$ and $ct_0^{(i_0)} \xrightarrow{\mathsf{rk}^{f_1}_{i_0 \to i_1}} ct_1^{(i_1)} \xrightarrow{\mathsf{rk}^{f_2}_{i_1 \to i_2}} \cdots \xrightarrow{\mathsf{rk}^{f_L}_{i_{L-1} \to i_L}} ct_L^{(i_L)}$,
- (Case $\beta = 1$) *or* a set of ciphertexts generated by the encryption algorithms independently, namely $ct_j^{(i_j)} \leftarrow_\$ \mathsf{Enc}(pk^{(i_j)}, m_j, j)$ for $j \in [0, L]$.

$\mathcal{A}$ aims to guess which case it is.

To prevent trivial attacks from $\mathcal{A}$, we also keep track of two sets $\mathcal{Q}_c$ and $\mathcal{Q}_{rk}$ to record the corrupted users and the tuples (i, j) that $\mathcal{A}$ obtains a re-encryption key $\mathsf{rk}^f_{i \to j}$, respectively. Based on that, there are two trivial attacks **TA1$''$-TA2$''$** to obtain information about the plaintexts underlying the challenge ciphertexts $\{ct_j^{(i_j)}\}_{j \in [0,L]}$.

TA1$''$: $\exists\, j \in [0, L]$, s.t. $i_j \in \mathcal{Q}_c$, i.e., $\mathcal{A}$ ever corrupts a challenge user i_j and obtains its secret key $sk^{(i_j)}$. In this case, $\mathcal{A}$ can decrypt the challenge ciphertext $ct_j^{(i_j)}$ with $sk^{(i_j)}$ and learn the underlying plaintext.

TA2$''$: $\exists\, i^* \in \mathcal{Q}_u$ and $\exists\, (i^*, j_1), (j_1, j_2), \ldots, (j_{t-1}, j_t) \in \mathcal{Q}_{rk}$ s.t. $j_t \in \mathcal{Q}_c$ for some $t \geq 1$, i.e., $\mathcal{A}$ gets a chain of re-encryption keys $\mathsf{rk}^{f'_1}_{i^* \to j_1}, \mathsf{rk}^{f'_2}_{j_1 \to j_2}, \ldots, \mathsf{rk}^{f'_t}_{j_{t-1} \to j_t}$ starting from some challenge user i^* and ending at some corrupted user j_t that $\mathcal{A}$ ever obtains its secret key $sk^{(j_t)}$. In this case, $\mathcal{A}$ can re-encrypt the challenge ciphertext $ct_v^{(i^*)}$ corresponding to user i^* via $ct_v^{(i^*)} \xrightarrow{\mathsf{rk}^{f'_1}_{i^* \to j_1}} ct_{v+1}^{(j_1)} \xrightarrow{\mathsf{rk}^{f'_2}_{j_1 \to j_2}} \cdots \xrightarrow{\mathsf{rk}^{f'_t}_{j_{t-1} \to j_t}} ct_{v+t}^{(j_t)}$, then decrypt $ct_{v+t}^{(j_t)}$ with $sk^{(j_t)}$ to learn information about the plaintext contained in the challenge ciphertext $ct_v^{(i^*)}$. This kind of trivial attacks is checked by the algorithm $\mathsf{CheckTA}$ defined in Fig. 8 throughout the experiment.

As such, we exclude the above two trivial attacks in the CUL experiment.

Below we show that the CUL security is implied by the SH security together with the CPA security for multi-hop FPRE.

Lemma 11 (SH + CPA $\Rightarrow$ CUL for Multi-Hop FPRE). *If a multi-hop FPRE scheme* mFPRE *is both* SH *and* CPA *secure, then it is also* CUL *secure. More precisely, for any PPT adversary* $\mathcal{A}$ *against the* CUL *security, there exist PPT algorithm* $\mathcal{B}$ *and* $\mathcal{B}'$ *s.t.*

$$\mathsf{Adv}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}(\lambda) \leq 2\mathfrak{n}^2 L \cdot \mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda) + 2(L+1) \cdot \mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B}'}(\lambda).$$

Proof of Lemma 11. We prove the theorem via a sequence of games G_0-G_3, where G_0 is the CUL experiment, and in G_3, $\mathcal{A}$ has a negligible advantage.

Game G_0: This is the CUL experiment (cf. Fig. 8). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}} = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Game $\mathsf{G}_{0.t}, t \in [0, L]$: It is the same as G_0, except that in the case of $\beta = 0$, the challenger generates the first t challenge ciphertexts $\{ct_j^{(i_j)}\}_{j \in [t]}$ by invoking

$ct_j^{(i_j)} \leftarrow_{\$} \mathsf{Enc}(pk^{(j)}, f_j(\cdots f_1(m)), j)$ independently for each $j \in [t]$, instead of generating them by re-encryption $ct_j^{(i_j)} \leftarrow_{\$} \mathsf{FReEnc}(\mathsf{rk}^{f_j}_{i_{j-1}\to i_j}, ct^{(i_{j-1})}, j-1)$.

Clearly, $\mathsf{G}_{0.0}$ is identical to G_0. Thus, we have $\Pr_0[\mathsf{Win}] = \Pr_{0.0}[\mathsf{Win}]$.

Below we show the computational indistinguishability between $\mathsf{G}_{0.t-1}$ and $\mathsf{G}_{0.t}$ based on the SH security.

Claim 7. For all $t \in [L]$, $|\Pr_{0.t-1}[\mathsf{Win}] - \Pr_{0.t}[\mathsf{Win}]| \le 2\mathfrak{n}^2 \cdot \mathsf{Adv}^{\mathsf{SH}}_{\mathsf{mFPRE},\mathcal{B}}(\lambda)$.

Proof. We construct a PPT algorithm $\mathcal{B}$ to break the SH security of mFPRE by simulating $\mathsf{G}_{0.t-1}/\mathsf{G}_{0.t}$ for $\mathcal{A}$ as follows.

Algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ is given the public keys $(pk^{(0)}_{\mathsf{SH}}, sk^{(0)}_{\mathsf{SH}}, pk^{(1)}_{\mathsf{SH}}, sk^{(1)}_{\mathsf{SH}})$ from its own challenger and has access to its own oracles $\mathcal{O}_{\text{ReKey}}, \mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Chal}}$. $\mathcal{B}$ wants to guess the challenge bit β_{SH} chosen by its own challenger.

$\mathcal{B}$ is constructed by simulating $\mathsf{G}_{0.t-1}/\mathsf{G}_{0.t}$ for $\mathcal{A}$ as follows. $\mathcal{B}$ chooses two distinct user indices $i'_{t-1}, i'_t \leftarrow_{\$} [\mathfrak{n}]$ uniformly at random. For user i'_{t-1} and user i'_t, $\mathcal{B}$ sets $pk^{(i'_{t-1})} := pk^{(0)}_{\mathsf{SH}}, sk^{(i'_{t-1})} := sk^{(0)}_{\mathsf{SH}}$ and $pk^{(i'_t)} := pk^{(1)}_{\mathsf{SH}}, sk^{(i'_t)} := sk^{(1)}_{\mathsf{SH}}$. For all other users $i \in [\mathfrak{n}] \setminus \{i'_{t-1}, i'_t\}$, $\mathcal{B}$ invokes KGen honestly to generate $(pk^{(i)}, sk^{(i)})$. Note that $\mathcal{B}$ owns the secret keys $\{sk^{(i)}\}_{i\in[\mathfrak{n}]}$ of all users.

(1) $\mathcal{B}$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, \mathcal{Q}_u = \emptyset$ and sends $\{pk^{(i)}\}_{i\in[\mathfrak{n}]}$ to $\mathcal{A}$.

- On receiving a re-encryption key query (i, j, f) from $\mathcal{A}$, $\mathcal{B}$ checks $\mathcal{A}$'s trivial attacks by checking if $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$, just like $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise $\mathcal{B}$ adds (i,j) to $\mathcal{Q}_{rk}$ and
  - if $i = i'_{t-1}$ and $j = i'_t$, $\mathcal{B}$ queries $\mathcal{O}_{\text{ReKey}}(f)$ to its own challenger and returns $\mathsf{rk}^f_{i'_{t-1}\to i'_t}$ it receives to $\mathcal{A}$;
  - otherwise, $\mathcal{B}$ answers $\mathcal{A}$'s query honestly by invoking $\mathsf{rk}^f_{i\to j} \leftarrow_{\$} \mathsf{FReKGen}(pk^{(i)}, sk^{(i)}, pk^{(j)}, f)$ using the secret key $sk^{(i)}$.
- On receiving a corruption query i from $\mathcal{A}$, $\mathcal{B}$ checks $\mathcal{A}$'s trivial attacks by checking if $(i \in \mathcal{Q}_u)$ or $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c \cup \{i\}) = 1$, just like $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$, otherwise $\mathcal{B}$ adds i to $\mathcal{Q}_c$ and returns $sk^{(i)}$ to $\mathcal{A}$.
- On receiving the challenge tuple $(\{i_j\}_{j\in[0,L]}, \binom{(\{f_j\}_{j\in[L]}, m)}{(m_0, m_1, \ldots, m_L)})$ from $\mathcal{A}$, $\mathcal{B}$ defines $\mathcal{Q}_u := \{i_j\}_{j\in[0,L]}$ and checks $\mathcal{A}$'s trivial attacks by checking if $(\exists j \in [0,L]$ s.t. $i_j \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$. If trivial attacks occur, $\mathcal{B}$ aborts the game with $\mathcal{A}$ and returns a random bit $\beta'_{\mathsf{SH}} \leftarrow_{\$} \{0,1\}$ to its own challenger. If $i_{t-1} \ne i'_{t-1}$ or $i_t \ne i'_t$, $\mathcal{B}$ also aborts the game and returns a random bit $\beta'_{\mathsf{SH}} \leftarrow_{\$} \{0,1\}$ to its own challenger. Otherwise, $\mathcal{B}$ chooses a random bit $\beta \leftarrow_{\$} \{0,1\}$ for $\mathcal{A}$. If $\beta = 1$, $\mathcal{B}$ answers $\mathcal{A}$ in the same way as in $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If $\beta = 0$, $\mathcal{B}$ generates the challenge ciphertexts $\{ct_j^{(i_j)}\}_{j\in[0,L]}$ as follows and returns them to $\mathcal{A}$.
  - For $j = 0$, $\mathcal{B}$ generates $ct_0^{(i_0)}$ by invoking $ct_0^{(i_0)} \leftarrow_{\$} \mathsf{Enc}(pk^{(i_0)}, m, 0)$.
  - For $1 \le j < t-1$, $\mathcal{B}$ generates $ct_j^{(i_j)}$ by invoking $ct_j^{(i_j)} \leftarrow_{\$} \mathsf{Enc}(pk^{(i_j)}, f_j(\cdots f_1(m)), j)$.

- For $j = t-1$, $\mathcal{B}$ queries $\mathcal{O}_{\textsc{Enc}}(f_{t-1}(\cdots f_1(m)), t-1)$ to its own challenger and receives (ctr, ct). $\mathcal{B}$ sets $ct_{t-1}^{(i_{t-1})} := ct$.
- For $j = t$, $\mathcal{B}$ queries $\mathcal{O}_{\textsc{Chal}}(\mathsf{ctr}, f_t)$ to its own challenger and receives ct'. $\mathcal{B}$ sets $ct_t^{(i_t)} := ct'$.
- For $t < j \leq L$, $\mathcal{B}$ generates $ct_j^{(i_j)}$ in the same way as in $\mathsf{Exp}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}^{\mathsf{CUL}}$.

(2) Finally, $\mathcal{B}$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}$ outputs $\beta'_{\mathsf{SH}} = 1$ to its own challenger if $\beta' = \beta$.

Let T denote the event that $i'_{t-1} = i_{t-1}$ and $i'_t = i_t$. Note that the values of i'_{t-1} and i'_t are completely hidden to $\mathcal{A}$ before $\mathcal{A}$ submits the challenge tuple $(\{i_j\}_{j\in[0,L]}, \cdots)$. Consequently, we have $\Pr[\mathsf{T}] = 1/\mathfrak{n}(\mathfrak{n}-1) \geq 1/\mathfrak{n}^2$.

Now we analyze the advantage of $\mathcal{B}$. If T does not occur, i.e., $i_{t-1} \neq i'_{t-1}$ or $i_t \neq i'_t$, $\beta'_{\mathsf{SH}} \leftarrow_{\$} \{0,1\}$ is randomly chosen by $\mathcal{B}$, and in particular, independent of β_{SH}. If T occurs, it is not hard to see that $\mathcal{B}$ simulates $\mathsf{G}_{0.t-1}$ perfectly for $\mathcal{A}$ in the case $\beta_{\mathsf{SH}} = 0$ and simulates $\mathsf{G}_{0.t}$ perfectly for $\mathcal{A}$ in the case $\beta_{\mathsf{SH}} = 1$. Consequently, we have that

$$\begin{aligned}
&\mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}}^{\mathsf{SH}}(\lambda) = |\Pr[\beta'_{\mathsf{SH}} = \beta_{\mathsf{SH}}] - \tfrac{1}{2}| \\
&= \tfrac{1}{2} \cdot |\Pr[\beta'_{\mathsf{SH}} = 1 \mid \beta_{\mathsf{SH}} = 0] - \Pr[\beta'_{\mathsf{SH}} = 1 \mid \beta_{\mathsf{SH}} = 1]| \\
&= \tfrac{1}{2}\Pr[\mathsf{T}] \cdot |\Pr[\beta' = \beta \mid \beta_{\mathsf{SH}} = 0 \wedge \mathsf{T}] - \Pr[\beta' = \beta \mid \beta_{\mathsf{SH}} = 1 \wedge \mathsf{T}]| \\
&= \tfrac{1}{2}\Pr[\mathsf{T}] \cdot |\Pr_{0.t-1}[\mathsf{Win}] - \Pr_{0.t}[\mathsf{Win}]| \\
&\geq \tfrac{1}{2\mathfrak{n}^2} \cdot |\Pr_{0.t-1}[\mathsf{Win}] - \Pr_{0.t}[\mathsf{Win}]|.
\end{aligned}$$

This completes the proof of Claim 7. ▌

<u>**Game** G_1:</u> It is the same as G_0, except that in the case of $\beta = 0$, the challenger generates all challenge ciphertexts $\{ct_j^{(i_j)}\}_{j\in[L]}$ (except $ct_0^{(i_0)}$) by invoking $ct_j^{(i_j)} \leftarrow_{\$} \mathsf{Enc}(pk^{(j)}, f_j(\cdots f_1(m)), j)$ independently for all $j \in [L]$.

Clearly $\mathsf{G}_1 = \mathsf{G}_{0.L}$ and by Claim 7, we have

$$|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]| \leq 2\mathfrak{n}^2 L \cdot \mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}}^{\mathsf{SH}}(\lambda).$$

<u>**Game** $\mathsf{G}_{1.t}, t \in [0, L]$:</u> It is the same as G_1, except that in the case of $\beta = 0$, the challenger generates the first t challenge ciphertexts $\{ct_j^{(i_j)}\}_{j\in[t]}$ by encrypting m_j, i.e., $ct_j^{(i_j)} \leftarrow_{\$} \mathsf{Enc}(pk^{(j)}, m_j, j)$, instead of encrypting $f_j(\cdots f_1(m))$, where $j \in [t]$.

Clearly, $\mathsf{G}_{1.0}$ is identical to G_1. Thus, we have $\Pr_1[\mathsf{Win}] = \Pr_{1.0}[\mathsf{Win}]$.

Below we show the computational indistinguishability between $\mathsf{G}_{1.t-1}$ and $\mathsf{G}_{1.t}$ based on the CPA security.

Claim 8. For all $t \in [L]$, $|\Pr_{1.t-1}[\mathsf{Win}] - \Pr_{1.t}[\mathsf{Win}]| \leq 2 \cdot \mathsf{Adv}_{\mathsf{mFPRE},\mathcal{B}'}^{\mathsf{CPA}}(\lambda)$.

Proof. We construct a PPT algorithm $\mathcal{B}'$ to break the CPA security of mFPRE by simulating $\mathsf{G}_{1.t-1}/\mathsf{G}_{1.t}$ for $\mathcal{A}$ as follows.

Algorithm $\mathcal{B}'$. Algorithm $\mathcal{B}'$ is given the public keys $\{pk^{(i)}\}_{i\in[\mathfrak{n}]}$ from its own challenger and has access to its own oracles $\mathcal{O}_{\text{ReKey}}, \mathcal{O}_{\text{Cor}}$. $\mathcal{B}'$ wants to guess the challenge bit β_{CPA} chosen by its own challenger.

(1) $\mathcal{B}'$ initializes $\mathcal{Q}_{rk} = \emptyset, \mathcal{Q}_c = \emptyset, \mathcal{Q}_u = \emptyset$ and sends $\{pk^{(i)}\}_{i\in[\mathfrak{n}]}$ to $\mathcal{A}$.
  - On receiving a re-encryption key query (i, j, f) from $\mathcal{A}$, $\mathcal{B}'$ checks $\mathcal{A}$'s trivial attacks by checking if $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk} \cup \{(i,j)\}, \mathcal{Q}_c) = 1$, just like $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}'$ returns $\perp$ to $\mathcal{A}$, otherwise $\mathcal{B}'$ adds (i, j) to $\mathcal{Q}_{rk}$ and queries (i, j, f) to its own oracle $\mathcal{O}_{\text{ReKey}}$. On receiving $\mathsf{rk}^{f}_{i\to j}$ from $\mathcal{O}_{\text{ReKey}}(i, j, f)$, $\mathcal{B}'$ returns $\mathsf{rk}^{f}_{i\to j}$ to $\mathcal{A}$.
  - On receiving a corruption query i from $\mathcal{A}$, $\mathcal{B}'$ checks $\mathcal{A}$'s trivial attacks by checking if $(i \in \mathcal{Q}_u)$ or $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c\cup\{i\}) = 1$, just like $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If trivial attacks occur, $\mathcal{B}'$ returns $\perp$ to $\mathcal{A}$, otherwise $\mathcal{B}'$ adds i to $\mathcal{Q}_c$ and queries i to its own oracle $\mathcal{O}_{\text{Cor}}$. On receiving $sk^{(i)}$ from $\mathcal{O}_{\text{Cor}}(i)$, $\mathcal{B}'$ returns $sk^{(i)}$ to $\mathcal{A}$.
  - On receiving the challenge tuple $(\{i_j\}_{j\in[0,L]}, \binom{(\{f_j\}_{j\in[L]}, m)}{(m_0, m_1, \ldots, m_L)})$ from $\mathcal{A}$, $\mathcal{B}'$ defines $\mathcal{Q}_u := \{i_j\}_{j\in[0,L]}$ and checks $\mathcal{A}$'s trivial attacks by checking if $(\exists j \in [0, L]$ s.t. $i_j \in \mathcal{Q}_c)$ or $\mathsf{CheckTA}(\mathcal{Q}_u, \mathcal{Q}_{rk}, \mathcal{Q}_c) = 1$. If trivial attacks occur, $\mathcal{B}'$ aborts the game with $\mathcal{A}$ and returns a random bit $\beta'_{\mathsf{CPA}} \leftarrow_{\$} \{0, 1\}$ to its own challenger, otherwise, $\mathcal{B}'$ chooses a random bit $\beta \leftarrow_{\$} \{0, 1\}$ for $\mathcal{A}$. If $\beta = 1$, $\mathcal{B}'$ answers $\mathcal{A}$ in the same way as in $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. If $\beta = 0$, $\mathcal{B}'$ generates the challenge ciphertexts $\{ct^{(i_j)}_j\}_{j\in[0,L]}$ as follows and returns them to $\mathcal{A}$.
    - For $j = 0$, $\mathcal{B}'$ generates $ct^{(i_0)}_0$ by invoking $ct^{(i_0)}_0 \leftarrow_{\$} \mathsf{Enc}(pk^{(i_0)}, m, 0)$.
    - For each $1 \le j \le t-1$, $\mathcal{B}'$ generates $ct^{(i_j)}_j$ by invoking $ct^{(i_j)}_j \leftarrow_{\$} \mathsf{Enc}(pk^{(i_j)}, m_j, j)$.
    - For $j = t$, $\mathcal{B}'$ sends $(i_t, m'_t := f_t(\cdots f_1(m)), m_t, t)$ to its own challenger and receives ct^*. $\mathcal{B}'$ sets $ct^{(i_t)}_t := ct^*$.
    - For $t < j \le L$, $\mathcal{B}'$ generates $ct^{(i_j)}_j$ in the same way as in $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$.

(2) Finally, $\mathcal{B}'$ receives a bit β' from $\mathcal{A}$, and $\mathcal{B}'$ outputs $\beta'_{\mathsf{CPA}} = 1$ to its own challenger if $\beta' = \beta$.

In the simulation, if $\mathcal{A}$ implements trivial attacks **TA1′-TA2′**, $\mathcal{B}'$ will abort the experiment, just like $\mathsf{Exp}^{\mathsf{CUL}}_{\mathsf{mFPRE},\mathcal{A},\mathfrak{n}}$. Otherwise, no trivial attacks from $\mathcal{A}$ implies that $i_t \notin \mathcal{Q}_c$ and there does not exist a chain of re-encryption keys from i_t to $j \in \mathcal{Q}_c$, while $\mathcal{Q}_c$ is also the corrupted users set for $\mathcal{B}'$'s challenger. Thus, $\mathcal{B}'$ never issue queries leading to trivial attacks **TA1** and **TA2**. So $\mathcal{B}'$ is able to simulate the game for $\mathcal{A}$.

Now we analyze the advantage of $\mathcal{B}'$. If $\beta_{\mathsf{CPA}} = 0$, then ct^* is an encryption of $m'_t = f_t(\cdots f_1(m))$, and thus $\mathcal{B}'$ simulates $\mathsf{G}_{1.t-1}$ perfectly for $\mathcal{A}$. If $\beta_{\mathsf{CPA}} = 1$, then ct^* is an encryption of m_t, and thus $\mathcal{B}'$ simulates $\mathsf{G}_{1.t}$ perfectly for $\mathcal{A}$. Consequently, we have that

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B}'}(\lambda) = |\Pr[\beta'_{\mathsf{CPA}} = \beta_{\mathsf{CPA}}] - \tfrac{1}{2}|\\
&= \tfrac{1}{2} \cdot |\Pr[\beta'_{\mathsf{CPA}} = 1 \mid \beta_{\mathsf{CPA}} = 0] - \Pr[\beta'_{\mathsf{CPA}} = 1 \mid \beta_{\mathsf{CPA}} = 1]|\\
&= \tfrac{1}{2} \cdot |\Pr[\beta' = \beta \mid \beta_{\mathsf{CPA}} = 0] - \Pr[\beta' = \beta \mid \beta_{\mathsf{CPA}} = 1]|\\
&= \tfrac{1}{2} \cdot |\Pr_{1.t-1}[\mathsf{Win}] - \Pr_{1.t}[\mathsf{Win}]|.
\end{aligned}$$

This completes the proof of Claim 8. ▌

<u>**Game G_2**</u>. It is the same as G_1, except that in the case of $\beta = 0$, the challenger generates all challenge ciphertexts $\{ct_j^{(i_j)}\}_{j\in[L]}$ (except $ct_0^{(i_0)}$) by encrypting m_j, i.e., $ct_j^{(i_j)} \leftarrow_\$ \mathsf{Enc}(pk^{(j)}, m_j, j)$ for all $j \in [L]$.

Clearly, $\mathsf{G}_2 = \mathsf{G}_{1.L}$ and by Claim 8, we have

$$|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]| \leq 2L \cdot \mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B}'}(\lambda).$$

<u>**Game G_3**</u>. It is the same as G_2, except that in the case of $\beta = 0$, the challenger generates the 0-th challenge ciphertext $ct_0^{(i_0)}$ by encrypting m_0, i.e., $ct_0^{(i_0)} \leftarrow_\$ \mathsf{Enc}($ $pk^{(0)}, m_0, 0)$, instead of encryption m.

Similar to Claim 8, we can show the computational indistinguishability between G_2 and G_3 based on the CPA security, and get that $|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{mFPRE},\mathcal{B}'}(\lambda)$.

Finally, note that in G_3, all challenge ciphertexts $\{ct_j^{(i_j)}\}_{j\in[0,L]}$ are always independently generated encryptions of m_j, i.e., $ct_j^{(i_j)} \leftarrow_\$ \mathsf{Enc}(pk^{(j)}, m_j, j)$ for all $j \in [0, L]$, regardless of the challenge bit β. Consequently, β is completely hidden to $\mathcal{A}$, and we have $\Pr_3[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Lemma 11 follows. □

Table of Contents